US Army Corps of Engineers

TECHNICAL REPORT GL-92-16

(2)

TWO-DIMENSIONAL PLANAR GEOSYSTEMS SUBJECTED TO THREE-DIMENSIONAL DYNAMIC LOADS

by

David W. Sykora

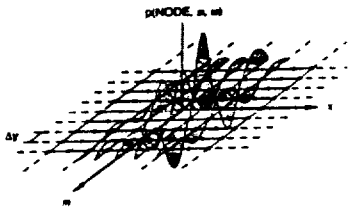
Geotechnical Laboratory

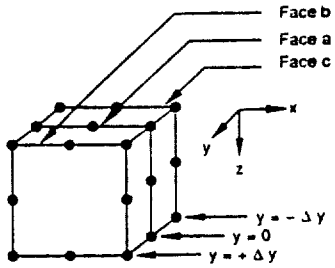
DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199

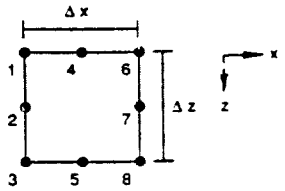
and

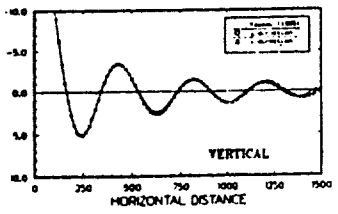
Jose M. Roesset

Department of Civil Engineering
University of Texas
Austin, Texas 78712-1076









September 1992
Final Report

Prepared for DEPARTMENT OF THE ARMY
Assistant Secretary of Army (R&D)
Washington, DC 20315

GEOTECHNICAL LABORATORY

The computer code provided in Appendix C is distributed "AS IS" and without warranty as to performance. Because of the many uses to which this code may be put and the variety of hardware used in conjunction with it, no warranty of fitness for a particular purpose is offered. While the developers have invested considerable time to create a high quality product, the user must assume the risk of using this software.

ABAQUS is a registered trademark of Hibbit, Karlsson and Sorensen, Inc., Providence, RI

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1992	Final report

4. TITLE AND SUBTITLE

Two-Dimensional Planar Geosystems Subjected to Three-Dimensional Dynamic Loads

5. FUNDING NUMBERS

Project No.
A91D-LR-002

6. AUTHOR(S)

David W. Sykora, Jose M. Roesset

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

USAE Waterways Experiment Station, Geotechnical Laboratory
3909 Halls Ferry Road, Vicksburg, MS 39180-6199 and
Department of Civil Engineering, University of Texas
Austin, TX 78712-1076

8. PERFORMING ORGANIZATION REPORT NUMBER

Technical Report
GL-92-16

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DEPARTMENT OF THE ARMY
Assistant Secretary of Army (R&D)
Washington, DC 20315

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report summarizes an analysis for estimating the variation of displacements in space and time produced by dynamic loads in complex isotropic media using a numerical approximation method. The important distinction is that the formulation allows certain three-dimensional (3-D) problems to be solved using a two-dimensional (2-D) numerical model. To implement this method, the stratigraphy and material properties cannot vary in a horizontal direction (2-D stratigraphy). However, the distribution and extent of loads may vary in both horizontal directions (3-D load) providing for the analysis of synthetic sources. These types of problems cannot be solved analytically but normally would be solved using a laborious 3-D numerical approximation.

The procedure involves condensing the 3-D dynamic stiffness matrices into equivalent 2-D matrices and representing the load function in the out-of-plane direction with a Fourier expansion. This report summarizes previous studies of wave propagation, the proposed formulation and computer implementation of a new finite element computer code, and the validation and parametric analysis of this code.

14. SUBJECT TERMS

Computer software
Dynamic displacements
Finite element method
Fourier analysis
Vibrations

15. NUMBER OF PAGES
194

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		

PREFACE

This study was conducted by the US Army Engineer Waterways Experiment Station (WES) for the Assistant Secretary of the Army (R&D), Project Number A91D-LR-002, as an In-House Laboratory Independent Research (ILIR) Program during FY 90 through FY 92. The title of the work unit was "Solution of Problems Involving Two-Dimensional Plane Continua Subjected to Three-Dimensional Dynamic Loads."

The methods of formulation used are an extension of the work done by Vincent Kang in his PhD thesis at the University of Texas at Austin. Dr. Kang solved the problem of dynamic displacements for a more simple class of problems -- horizontally layered pavement systems. His cooperation and assistance at the early stages of this work are greatly appreciated. During the course of this study, the results of another study using element condensation for pavement systems were reported by Hanazato et al. (1991).

This ILIR study was performed by Mr. David W. Sykora of the Earthquake Engineering and Seismology Branch (EESB), Earthquake Engineering and Geosciences Division (EEGD), Geotechnical Laboratory (GL), WES, under the technical advice of Prof. Jose M. Roesset, Department of Civil Engineering, The University of Texas at Austin (UT), Austin, Texas. The following engineers are acknowledged for their contributions, be it algorithms, ideas, suggestions, comments, or provocative conversations: Professors Kenneth H. Stokoe II, and John L. Tassoulas, UT, Mr. Ronald E. Wahl and Dr. John Peters, WES, and Mr. George Smith, Silicon Graphics, Inc. Messrs. Willie McGeehee and Daniel Habeeb, WES, created the figures in Parts I, II, and III.

The authors are grateful to Professor Eduardo Kausel, Massachusetts Institute of Technology, Cambridge, Massachusetts, for providing a copy of the computer code *PUNCH* used to conduct validation studies. Ms. Jennifer Davis, WES, assisted in generating outputs using this program.

Overall direction at WES was provided by Dr. A. G. Franklin, Chief, EEGD, and Dr. William F. Marcuson III, Director, GL.

At the time of publication of this report, Director of WES was Dr. Robert W. Whalin. Commander and Deputy Director was COL Leonard G. Hassel, EN.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont'd)

LIST OF FIGURES (cont'd)

TWO-DIMENSIONAL PLANAR GEOSYSTEMS SUBJECTED TO THREE-DIMENSIONAL DYNAMIC LOADS

PART I: INTRODUCTION

General

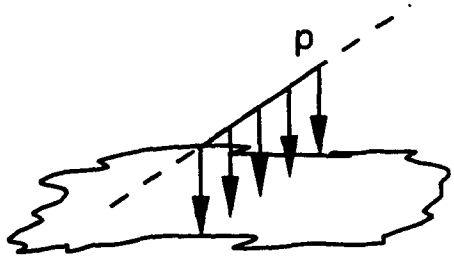
1. Many problems of Elastodynamics -- the study of the response of an elastic body to dynamic forces -- have been evaluated experimentally, solved explicitly, or have been properly formulated for implicit solution over a 160 year history dating back to Cauchy, Poisson, Stokes, and Lame' during the early to mid- 1800's. As each class of problems is solved, more complex problems are presented or more accurate, more efficient, or simpler means to solve a problem are desired. Accuracy, efficiency, and simplicity are all important aspects to the integration of new technology into military systems.

2. The present study is a systems analysis of the forward problem to estimate the variation of displacements in space and time produced by dynamic loads in complex isotropic media, consisting of dipping, discontinuous, and/or irregular layers, using a numerical approximation method. The distinguishing feature of this study is a formulation that allows three-dimensional (3-D) problems to be solved using a two-dimensional (2-D) numerical model. To implement this method, the stratigraphy and material properties of the model cannot vary in a horizontal direction (2-D stratigraphy). However, the distribution and extent of loads may vary in both horizontal directions (3-D load) providing for the analysis of synthetic vibratory sources such as a Vibroseis truck. Examples of 2-D and 3-D loads are shown in Figure 1. These types of problems cannot be solved analytically but normally would be solved using a laborious 3-D numerical approximation.

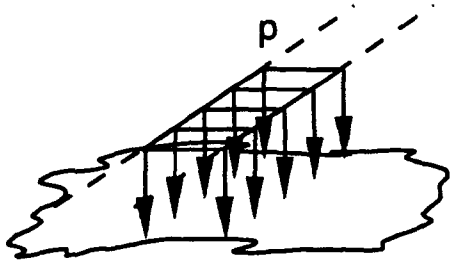
3. Soil dynamics studies conducted in the 1950's and 1960's using finite difference and finite element methods, and in the 1970's and 1980's using Green's functions and boundary element models, generally assumed plane harmonic waves and horizontally layered media extending to infinity. The subsurface distribution of materials at most sites is not simple nor is it conducive to analytical closed-form solutions of wave propagation problems.

TWO-DIMENSIONAL LOADS :

Line [F/L]

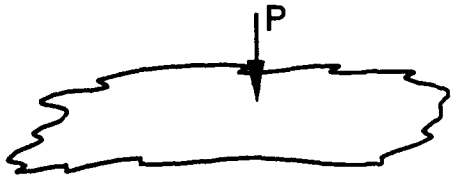


Strip [F / L / W]



THREE-DIMENSIONAL LOADS :

Point [F]



Rectangular [F/L^2]

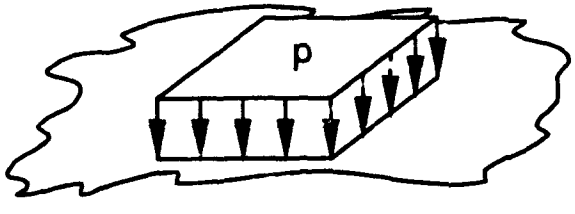


Figure 1. Comparison between 2-D and 3-D loads

Sloping strata of finite length, an irregular ground surface, and/or two-dimensional load distributions are prevalent. The present study describes a procedure to analyze wave propagation in these more complex systems.

4. This report describes a means to efficiently calculate dynamic vertical displacements by representing 3-D systems with an equivalent 2-D model. The finite element method was selected for computational solution to permit discretization of geosystems with numerous materials of arbitrary geometry. Initially, only steady-state dynamic loads are considered although the computer code can be easily adapted to allow the solution for transient loads by performing an additional Fourier transformation of the load function from the time to frequency domain.

5. The formulation involves two primary components: the condensation of 3-D dynamic stiffness matrices to equivalent 2-D matrices and the representation of the distribution of loads in the out-of-plane direction using a Fourier expansion. This strategy was explicitly proposed for axi-symmetric problems by Winnicki and Zienkiewicz (1979) and Lai and Booker (1991) and for 3-D formulations by Runesson and Booker (1982, 1983) and Lin and Tassoulas (1987). This strategy was used specifically for wave propagation studies in horizontally layered pavement systems by Kang (1990) and Hanazato et al. (1991). The 2-D system of equations are first solved in the frequency and wave-number domain; inverse Fourier transforms are then performed to obtain the solution as a function of out-of-plane distance and time, if so desired.

6. One objective of this study is to examine the potential for determining elastic moduli (i.e., shear, constrained, and Young's moduli) in complex systems of soil, rock, and structural materials from measured motions (the inverse problem). The Spectral-Analysis-of-Surface-Waves (SASW) method (Nazarian and Stokoe 1985a, 1985b) is one possible existing method of field measurement and mathematical inversion to determine the moduli of horizontally layered systems. This method involves the use of signal processing techniques on two measured vertical components of motion spaced at equal increments from the vibratory source. A similar procedure of determination is desired for more complex systems. In addition, the use of artificial neural networks holds promise to improve inversion schemes (Rix and Leipski 1991). Therefore, Rayleigh wave propagation will be of primary interest. Rayleigh waves normally contain most of the energy of wave propagation for the near

propagation for the near surface regime and Rayleigh wave energy will attenuate with distance at a much lower rate than body waves. The response at the ground surface is normally of greatest interest since it provides the easiest access for measurements.

Terminology

7. It is useful at this point to define some terminology. Geosystems are systems containing soil, rock, and possibly embedded foundations. Three-dimensional loads may be either point loads or loads acting over a finite area in plan (e.g., tires, tracks of vehicles, platen of Vibroseis truck, or blast). Two-dimensional loads are synonymous with plane waves or line loads extending to infinity in the direction perpendicular to the analysis plane. Plane waves refer to conditions where all points on a plane perpendicular to the direction of wave propagation undergo an identical incident disturbance at all instants of time during the disturbance. The term "irregular" applied to surfaces and layers is synonymous with the terms "dipping" (i.e., non-horizontal) or "discontinuous" (i.e., of finite length), or both, and includes layers with varying thickness. Contacts between layers can be approximated with a series of second degree parabolic segments.

Assumptions for Two-Dimensional Systems

8. A common assumption used to reduce the computational effort for the engineering analysis of stress and strain in boundary value problems of interest for geotechnical engineering applications is that of plane strain. Plane strain implies that the displacements in the direction perpendicular to a two-dimensional plane are equal to zero (Love 1944). This assumption reduces the scope of a problem from three to two dimensions. Conditions of plane strain require 2-D geometry and boundary conditions and loads that are uniform in the direction perpendicular to the plane under consideration (Timoshenko and Goodier 1970). A plane wave with particle motion only in the 2-D analysis plane is consistent with this assumption.

9. A surface load distributed over a finite surface induces stresses that vary in three principal directions. If stresses vary in a direction perpendicular to the analysis plane, displacements and strains will be non-

zero. Therefore, three-dimensional loads are inconsistent with plane strain assumptions. Axi-symmetric modeling is an alternative procedure for one-dimensional soil profiles. Many synthetic loads applied to the earth are of small dimensions relative to the extent of the analysis plane. For example, Rayleigh waves produced by a Vibroseis truck propagate through a layered half-space in three dimensions invalidate the assumptions of plane strain.

10. Examples of two-dimensional analysis planes and boundary conditions in systems that may exist in a state of plane strain include planes perpendicular to the axis of long, straight structures: tunnels (without rock-bolt reinforcement), embankments, retaining walls (without tiebacks or anchors), and vertical planes through isotropic soil deposits and geologic media that have no variation in profile for some arbitrary direction. Uniform loadings for these examples would include self weight (body forces) and hydrostatic (pool) loads for embankments, lateral forces on retaining walls, roof stresses in tunnels, and surcharges (e.g., highway embankment) in soil deposits or geologic media. Some examples are shown in Figure 2 and assume that the soil-structure system extends to a large distance relative to the predominant wavelength and distance from the source.

11. This study deals with the analysis of "planar" geosystems which proves to be beneficial from a computational standpoint. The primary assumptions are that the geometry and boundary conditions of the system and the distribution of material properties are planar (2-D) but the loads are non-planar (3-D). This set of conditions has a broader range of applications than that for plane strain while circumventing expensive 3-D solution methods.

Overview of Report

12. This report presents a broad discussion of aspects related to the analysis of Rayleigh wave propagation in geosystems using numerical methods. A review of previous studies is presented first. Next, the mathematical formulation and computer implementation of the finite element method, element condensation, and Fourier superposition are described. Validation and parametric analysis of the computer code are presented along with a comparison of computation times with 3-D finite element codes. The main part of the report concludes with a summary section. A listing of the primary computer program, *vib3*, and a sample problem are included in the appendices.

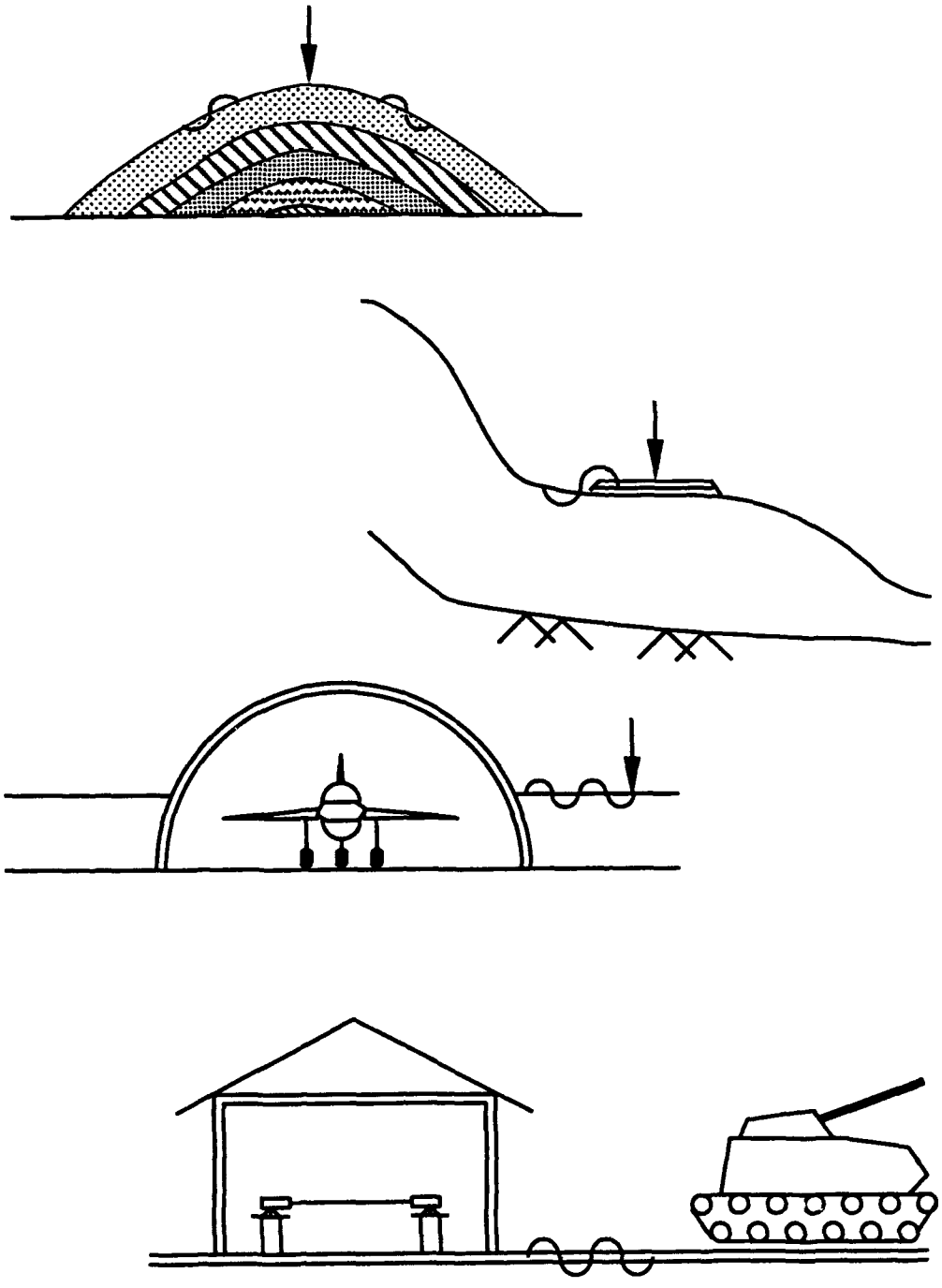


Figure 2. Examples of wave propagation problems in planar systems

PART II: EXISTING SOLUTIONS FOR DYNAMIC LOADS

Introduction

13. The evolution and the state of knowledge for dynamic loads acting on elastic media, particularly that involving coupled compression (P), in-plane shear (SV), and Rayleigh (R) waves, was reviewed to provide insight into which problems have been solved, what approaches were used, what conclusions have been reached, and which studies provide a proper basis for comparison or validation of the proposed formulation. Some general conclusions are:

a. Almost all the studies for layered systems considered plane wave propagation.

b. Many of the earlier studies that examined R-wave energy did not include in-plane P-SV waves,

c. Experimental studies generally focused on "thin plate" tests that have plane stress boundary conditions which are inconsistent with stress conditions for most geosystems, and

d. Plane strain conditions were generally assumed for theoretical and numerical studies.

The distribution of stresses caused by static point loads were also compared using closed-form solutions to quantify the errors associated with incorrectly modelling 2-D and 3-D loads. A presentation of these findings is made in Appendix A.

14. The literature reviewed has been categorized for purposes of explanation into four groups: fundamental studies (point loads in full or half-spaces), exact solutions for layered systems, experimental studies (laboratory and field measurements), and numerical and theoretical approximations.

Fundamental Studies

15. The study of dynamic displacements and wave propagation began in the early 19th century with Poisson and Kelvin. Stokes, Love, Rayleigh, and Lamb in the late 19th century further defined fundamental aspects of wave propagation in layered media and spheres. A summary of consequent studies that pertain to in-plane surface waves is presented in Table 1. Research studies as recent as Vardoulakis and Vrettos (1988) and Banerjee and Mamoon (1990) still consider the solution and formulation for three-dimensional

(point) load acting on or in an elastic half space. The studies listed in Table 1 are not directly applicable to the analyses and comparisons in this study because most deal with non-dispersive media or impulsive sources, or both. Many studies consider 3-D loads but the medium is too simple to use for validation.

Table 1

Initial Studies of In-Plane Surface Waves Produced by Dynamic Loads

Study	Load Distribution	Load Type	Solution
Mindlin (1936)	Internal point load	Harmonic	Exact integrals
Pinney (1954)	"	Impulsive (Pure P or S)	Exact integrals
Pekeris (1955)	Surficial point load	Impulsive	Contour integration
Mooney (1974)	Surficial point load	Impulsive	Elliptic integrals
Vardoulakis and Vrettos (1988)	Line load (planar)	Harmonic	Numerical solution eigenvalue problem

Closed-Form Solutions for Layered Systems

16. Numerous studies have been conducted since the 1950's to develop different means to solve wave propagation problems considering the wide range of conditions that would affect wave propagation. The response of horizontally layered media extending to infinity and overlying a half-space was first addressed by Thomson (1950) and corrected by Haskell (1953). Dunkin (1965) also added a correction to this formulation to maintain an acceptable degree of accuracy at high frequencies. Other studies since that time have considered refinements or have broadened the range of solvable conditions (e.g., Pestel and Leckie, 1963; Harkrider, 1964; Waas, 1972b; and Kausel and Roesset, 1981). Green's functions were evaluated numerically by Apsel (1979) and Kausel (1981). Studies utilizing closed-form solutions are listed in Table 2.

Table 2

Theoretical Solutions for Rayleigh Plane Wave Propagation in Layered Systems

Study	Load Distribution	Load Type	Approach
Thomson (1950) & Haskell (1953)	Plane wave	Harmonic	Transfer Matrix
Pestel and Leckie (1963)	"	"	"
Harkrider (1964)	Point or line load	Explosive, Point, or Strike-slip fault	Matrices to derive integrals
Dunkin (1965)	Plane wave	Harmonic	Transfer Matrix
Waas (1972b)	"	"	"
Apsel (1979)	Point or disk load	"	Green's functions
Kausel and Roesset (1981)	Plane wave	"	Stiffness Matrix
Kausel (1981)	Point, disk, or ring load	Harmonic	Green's functions

17. The studies lumped into this category represent a significant capability to calculate the dynamic displacements in an ideally layered system. These studies are still used successfully in various applications. Although the assumptions for an ideal layered system are unrealistic for many actual conditions, reasonable results can be obtained for simple soil and rock systems. The matrix solutions can be applied to plane wave propagation problems which are not of interest here. The Green's function solutions allow analysis of axi-symmetric problems which means 3-D loads (point and disk) in 2-D geometry (function of depth and radius).

18. The computer program developed as part of this study was validated using a solution method reported by Kausel (1981) for axi-symmetric problems with horizontally layered systems extending to infinity. The method by Kausel involves evaluating discrete Green's functions numerically and approaches the exact solution as the number of layers increases. More complicated problems cannot be solved accurately with this method. As a

consequence, researchers have had to extract information from experimental measurements and numerical and theoretical approximations.

Experimental Studies

19. Experimental studies, consisting of laboratory thin plate tests and field vibration tests, were predominant in the late 1950's and 1960's both in the laboratory and in the field. This thrust seems to correspond to a proliferation in the availability and use of electronic equipment and analog recording devices. This period is marked by the general absence of theoretical studies and generally pre-dates the solution of these problems via numerical techniques using computers.

20. Measured values provide realistic assessment of usefulness and applicability of a method but are not accurate and consistent enough for validation. A sufficient signal-to-noise ratio must be obtained and repeatable measurements must be available. Measurements made in the laboratory produce a wide range of possible values. Field measurements have inherent scatter and uncertainties because of the natural (unknown) variabilities of soil deposits. However, these same conditions must be recognized when proposing or fielding a new system for field measurement.

Laboratory studies

21. Several laboratory studies reported in the literature are listed in Table 3. Most laboratory studies examined the propagation of waves along the edge of thin plates. This type of test consists of cutting shapes from plates of material on the order of 0.16 cm in thickness, standing the plate on edge and placing receivers at various locations, producing an impulse across the thickness of the plate at some point, and then recording the wave as it propagates through the irregularity. A simple schematic of the system used by most researches in the laboratory is shown in Figure 3. Circular disks and concentric rings were used rather than quadrilateral shapes in one study. Some of the first studies of this type were reported by Kato and Takagi (1956) and Viktorov (1958) although little is known about the experimental system or the materials used.

22. The studies listed in Table 3 are subdivided into two categories: irregular surface (wedges) and irregular layers (step changes or material interfaces). The data for irregular surfaces is compared in a series of plots

Table 3

Comparison of Experimental Studies of Rayleigh Wave Propagation in Thin Plates

Study	Material	Thickness (cm)	ν	Wave Velocity (km/sec) P-wave	S-wave	R-wave	ρ (g/cc)
Irregular Surface:							
deBremaecker (1958)	Polystyrene	0.16	0.17	1.82	1.15	(1.04)*	
Viktorov (1958)	Dural						
Knopoff and Gangi (1960)	Aluminum	-	0.266	5.76	3.26	(3.00)	
Pilant, Knopoff & Schwab (1964)	Aluminum	0.18	0.25	5.4	3.12	2.88	
McGarr and Alsop (1967)	Polystyrene	0.16	(.24)	2.03	1.18	1.08	
Irregular Layers:							
Oliver, Press, & Ewing (1954)	Various**	0.16	0.21-0.28	1.96-5.58	1.18-3.25	1.08-2.99	1.22-9.03
Kuo & Thompson (1963)	Plexiglass	0.16	0.25	2.00	1.14	(1.05)	
	over Panelyte	0.16	0.27	3.20	1.80	(1.65)	
McGarr & Alsop (1967)	Plexiglass	0.16	(.35)	2.36	1.37	1.28	1.22
	abut Polystyrene	0.16	(.24)	2.03	1.18	1.08	1.08

* denotes estimated values (no specified by authors)
** Used twelve different material; ranges given

shown in Figures 4 through 7 along with a first-order-theory approximation (Hudson and Knopoff 1964). The measured coefficients for transmitted and reflected energy are compared in Figures 4 and 5, respectively. The phase shifts for transmitted and reflected Rayleigh wave energy reported by Pilant, Knopoff, and Schwab (1964) are shown in Figures 6 and 7 and compared with first-order theory (Hudson and Knopoff 1964).

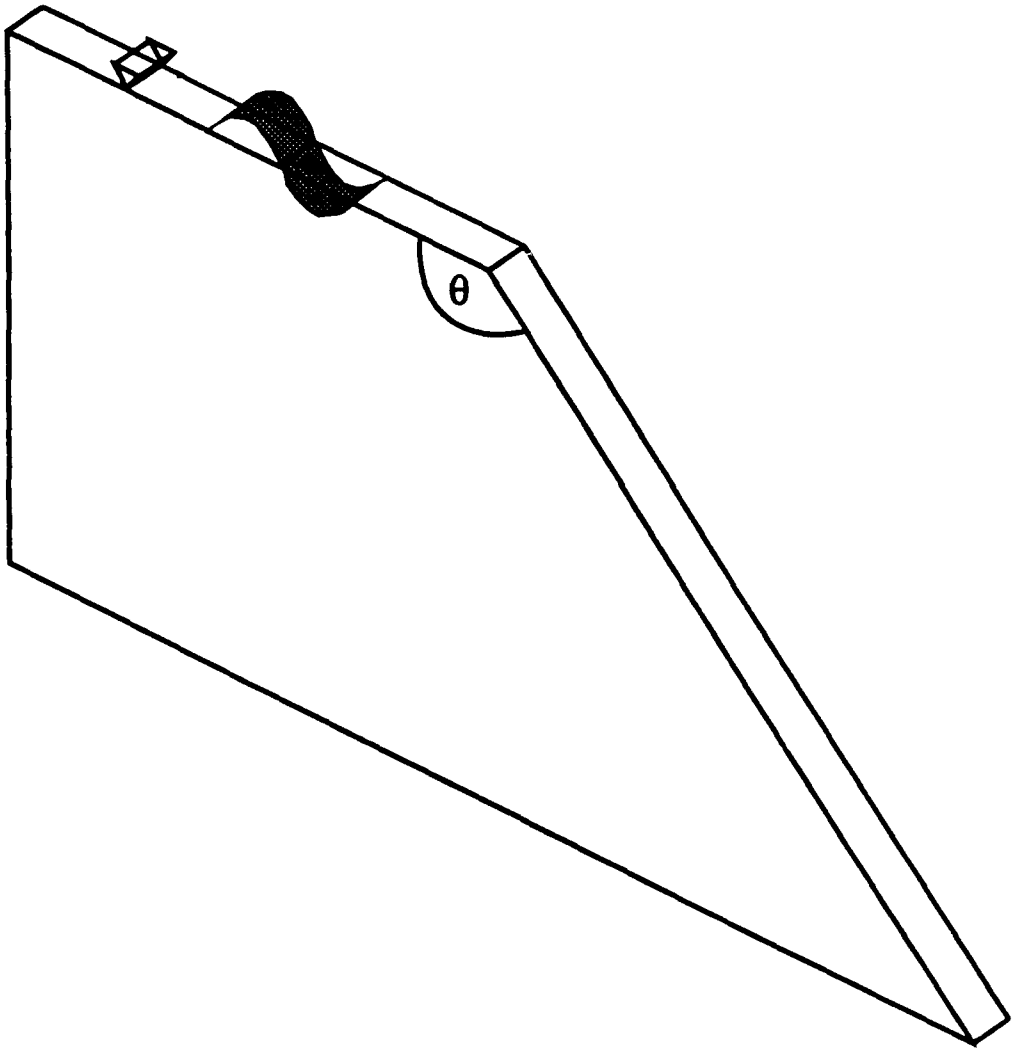


Figure 3. Schematic of "thin plate" test

23. The transmission coefficients for irregular surfaces from the four experimental studies are shown in Figure 4. In general, the measured values are considerably different although some general trends exist. The

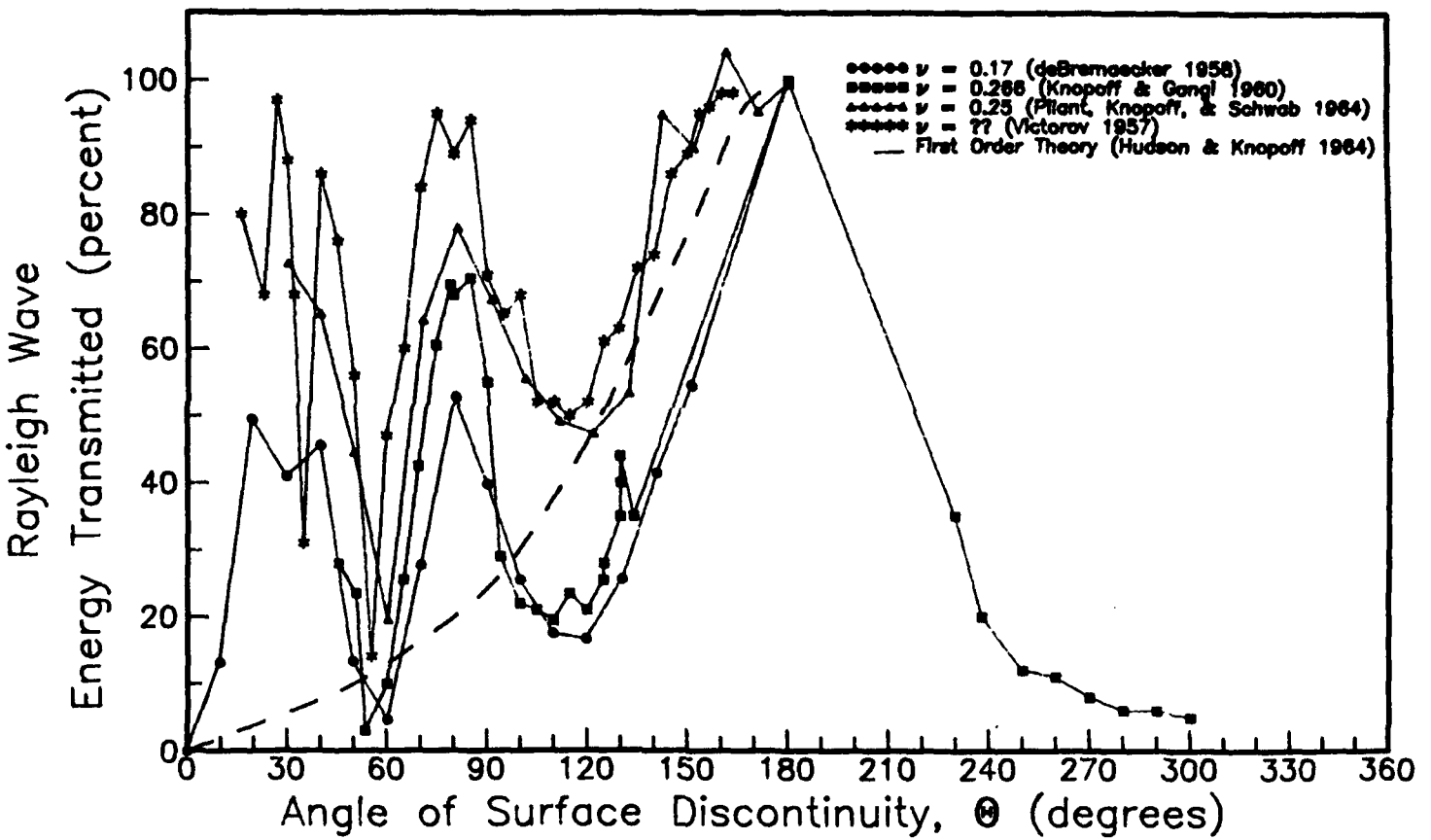


Figure 4. Results of early experimental surface wave tests showing variation of Rayleigh wave energy transmitted past discontinuity in surface

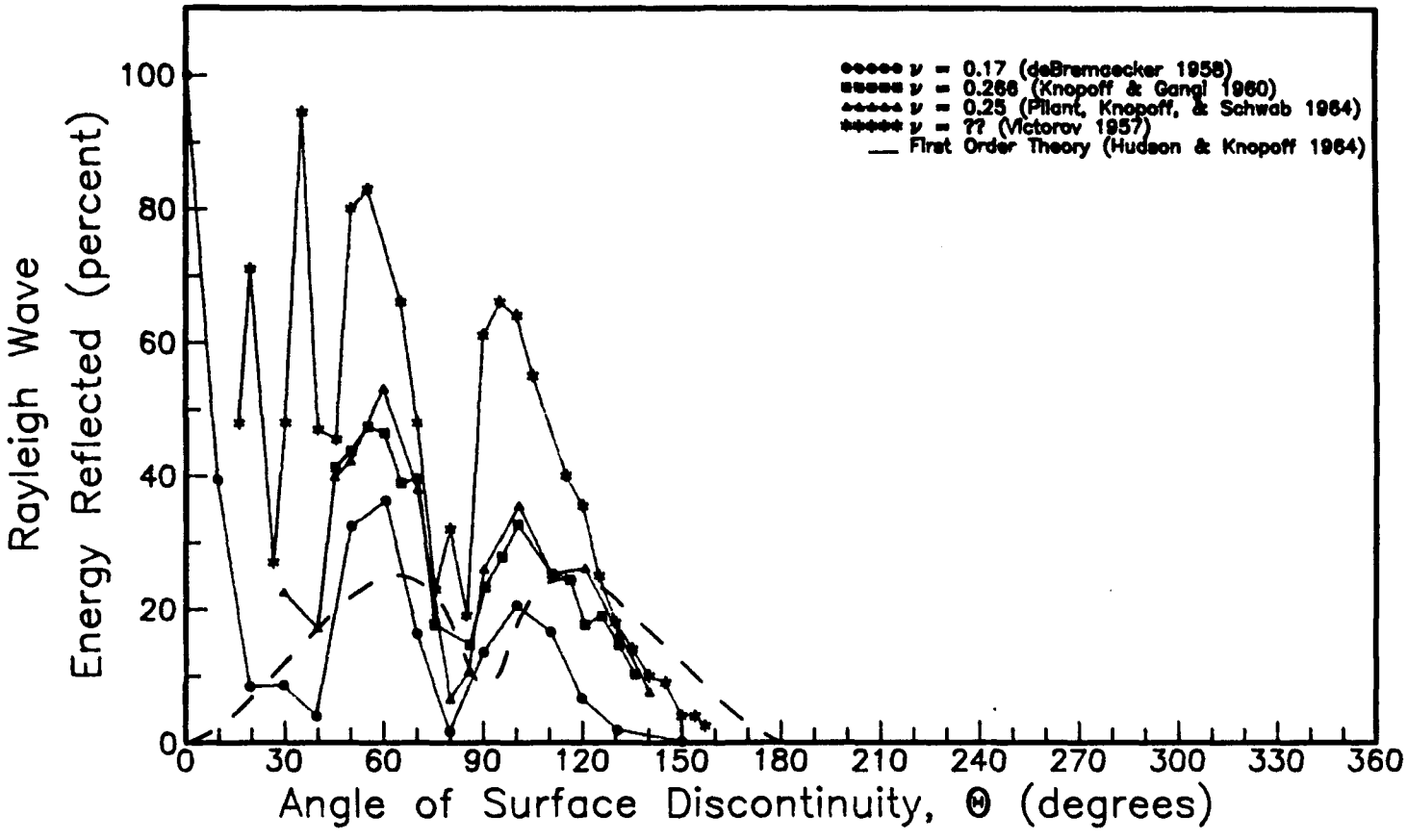


Figure 5. Results of early experimental surface wave tests showing variation of Rayleigh wave energy reflected past discontinuity in surface

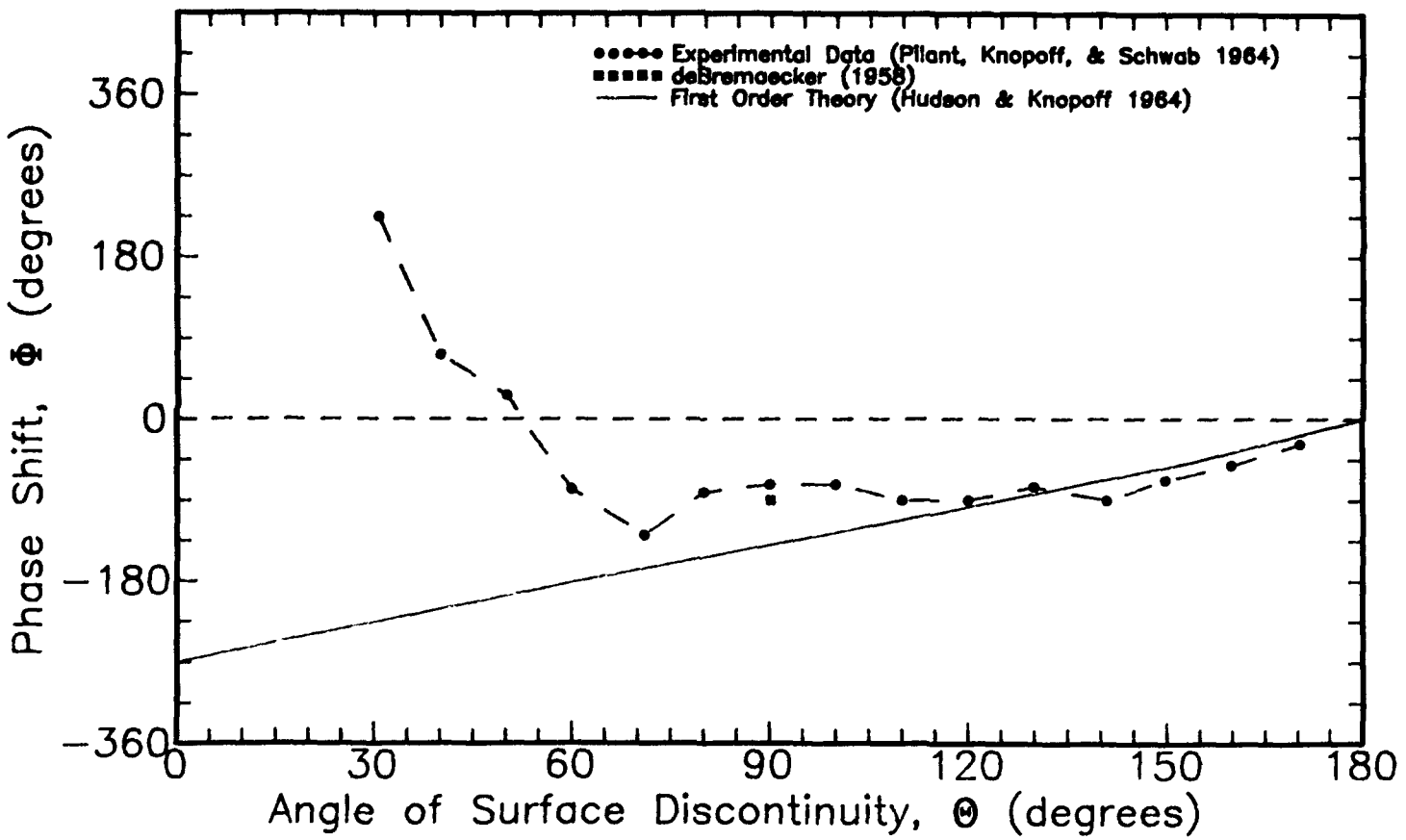


Figure 6. Results of early experimental surface wave tests showing variation of phase of Rayleigh waves transmitted past discontinuity in surface

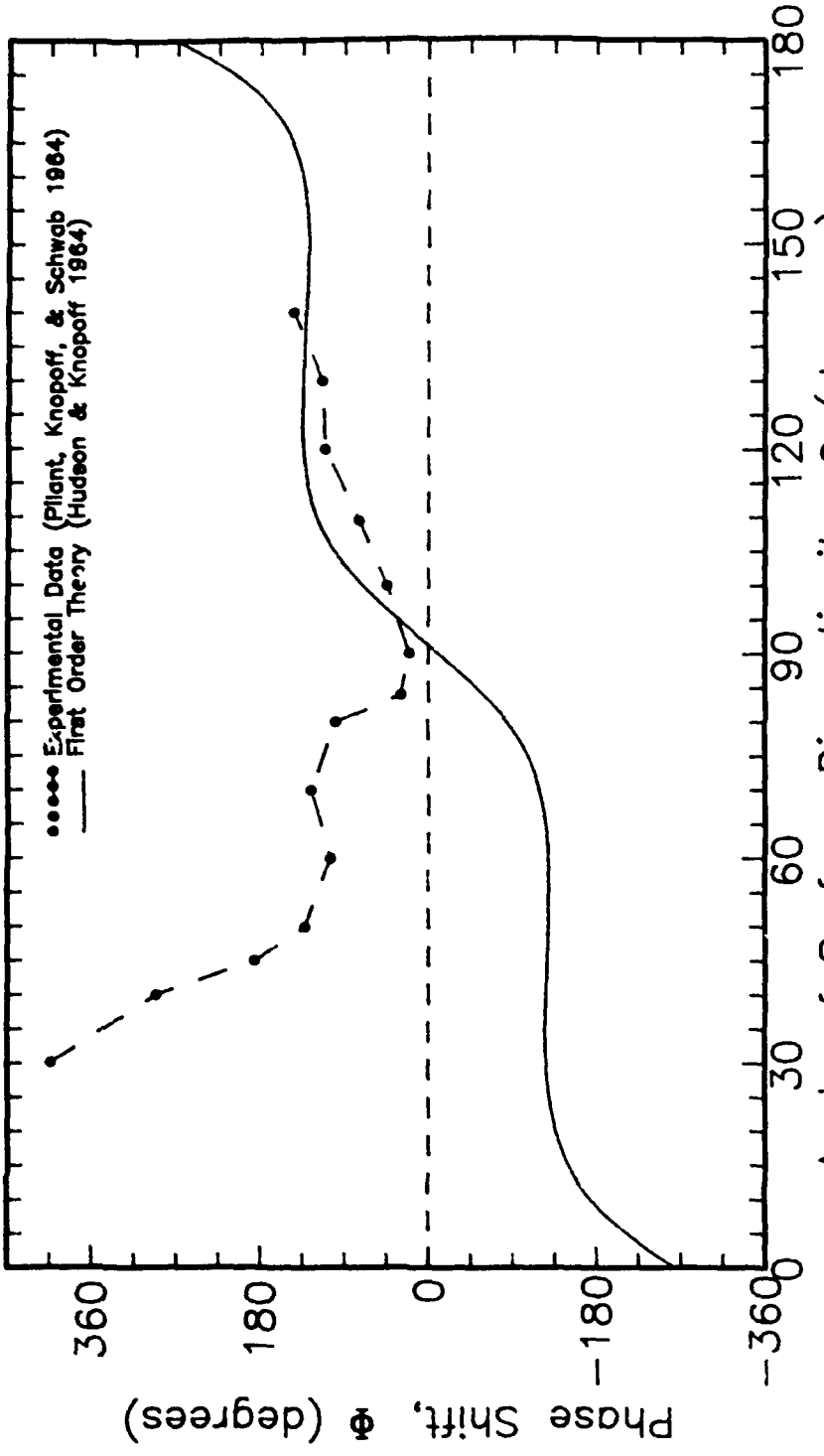


Figure 7. Results of early experimental surface wave tests showing variation of phase of Rayleigh waves reflected past discontinuity in surface

energy transmitted fluctuates considerably over the range of surface angles of 0 (free end) to 180 degrees (continuous horizontal surface). For troughs, the percentage of energy transmitted ranges from 2 to 20 at surface angles of 55 to 60 degrees and 15 to 50 at angles of 110 to 120 degrees (multiple of first set). Peaks occur at angles of 75 to 80 with percentages ranging from 50 to 95 and somewhere between 10 and 30 degrees. The pattern of peaks and troughs is relatively uniform, changing about every 30 degrees up to a surface angle of 120 degrees.

24. The reflection coefficients for irregular surfaces from the four experimental studies are shown in Figure 5. As expected, there is a strong correspondence between peak reflection coefficients and trough transmission coefficients in Figure 4 (angles of about 55 and 100 degrees). The peak energy reflected ranges between 35 and 82 percent at 50 to 60 degrees and 20 to 70 percent at 100 degrees.

25. The phase relationships for transmission and reflection presented by Pilant, Knopoff, and Schwab (1964) are shown in Figures 6 and 7. An observation by deBremaecker (1958) is also shown in Figure 6. The relationships for first-order theory by Hudson and Knopoff (1964) are also shown. The experimental values tend to show poor agreement with the first-order theory at surface angles less than 70 to 80 degrees.

Field studies

26. Field measurements focused on the propagation of surface waves through trenches in soil and rock. A few of these studies include: Barkan (1962), Dolling (1965 and 1970), and Woods (1968). The effect of trench dimensions normalized to the predominant wavelength, such as trench width, height, and length, were examined.

Numerical and Theoretical Approximations

27. The bulk of the studies that examine surface wave propagation in complex geosystems are numerical or theoretical approximations. Numerical approximations include the use of the finite difference (FD) method, the finite element (FE) method, the boundary element (BE) method, or a combination of these methods. Some of the early studies used wavefunction expansion to define body wave propagation in the media and then tried to solve for points along the boundary (an early form of BE). Other methods include: the ray path

method, the Aki-Larner method, and Alsop's method. Some of these studies are conveniently described and compared in general and specific terms by Knopoff (1969) and Yanovskaya (1989).

28. Theoretical approximations to determine reflection and transmission coefficients of surface waves can be categorized into three groups: Green's functions, vertical boundary approximations, and superposition of waves. Nearly all of the early studies used a Green's function solution (GF) which is a mathematical formulation of the Huygens-Fresnel principle. Some assumptions must be used in conjunction with this approach (thereby making it an approximation) because of the unknown stresses and displacements in complex media. For instance, Hudson and Knopoff (1964) neglected the reflected Rayleigh wave; Alsop (1966) assumed that body waves generated by surface waves impacting a surficial step change are small compared to the Rayleigh waves and could be neglected. Several others used this assumption to examine surface wave propagation through media with other geometric shapes. The number and severity of assumptions tended to decrease as the studies progressed chronologically.

29. The studies researched were divided into three categories for convenience. Studies examining the effects of irregular surfaces (e.g., step changes, vertical contacts, and wedges) on Rayleigh wave propagation are presented in Table 4. Studies examining the effects of canyons (e.g., empty and filled canyons with elliptical, circular, and arbitrary shapes) on Rayleigh wave propagation are presented in Table 5. Studies examining the effects of irregular subsurface layers (e.g., dipping layers and curved contacts) on Rayleigh wave propagation are presented in Table 6. General configurations for these categories are shown in Figure 8. This collection of studies allows for the analysis of most configurations of material geometry. However, the loads considered by nearly all of these studies are 2-D (plane waves). The problem of 3-D loads is more difficult to solve.

Table 4

Studies Examining Rayleigh Plane Wave Propagation

Through Irregular Surfaces for 2-D Geometries

Study	Case(s)	Approach
Kane & Spence (1963)	Wedge; wave from ∞	Iterate sol'n at boundaries
Hudson & Knopoff (1964 & 1967)	Wedge; source at apex	GF; neglected reflected waves
Mal & Knopoff (1965)	Step change	GF
Mal & Knopoff (1966)	Wedge; wave from ∞	GF
Lopez-Soto (1967)	(unknown)	GF; with body waves
McGarr & Alsop (1967)	Step change Discontinuous layers	GF; no body waves generated
Gutdeutsch (1969)	Wedge; source at apex	Empirical theory
Waas (1972a)	Trench	FE
Malischewski (1974 & 1976)	Vertical contact	GF; no body waves generated
Its & Yanovskaya (1977 & 1979)	Curved sub-vertical interface	GF
Segol, Lee, & Abel (1978)	Trench	FE
Lutikov (1979)	Vertical interface	GF
Fujii et al (1980)	Trench	FD
Fuyuki & Matsumoto (1980)	Trench	FD
Sanchez-Sesma (1983)	Ridge	BE
Fujii et al (1984)	Wedge	FD
Fuyuki & Nakano (1984)	Upward step change	FD
Ohtsuki & Yamahara (1984)	Wedge; source at ∞	FE/FD
Sanchez-Sesma, Chavez-Perez, & Aviles (1984)	3-D surface irreg.	BE
Gautesen (1985)	Right-angle wedge	Numerical integration
Sanchez-Sesma, Bravo, & Herrera (1985)	Topographic irreg.	BE
Sanchez-Sesma, Perez-Rocha, & Chavez-Perez (1985)	3-D surface irreg.	BE
Milder (1991)	Rough surface	Series of Helmholtz equations

Table 5

Studies Examining Rayleigh Plane Wave Propagation Through Canyons

Study	Case(s)	Approach
Lee (1978)	Hemi-spherical canyon	WFE
Bard & Bouchon (1980)	Bi-dimensional valleys	Aki-Larner method
Dravinski (1980)	Alluvial valley, arbitrary shape	(unknown)
Wong (1982)	Semi-elliptical & semi-circular canyons	BE
Lee & Langston (1983)	3-D circular basins*	Ray path
Sanchez-Sesma (1983)	3-D basins	BE
Lee (1984)	Hemi-spherical alluvial valley	BE
Kawase (1988)	Semi-circular canyon	BE
Eshraghi & Dravinski (1989a)	3-D canyons	BE
Khair, Datta, & Shah (1991)	Cylindrical alluvial valley	FE/BE

* Transient source

Table 6

Studies Examining Rayleigh Plane Wave Propagation

Through Dipping or Irregular Layers

Study	Case(s)	Approach
Kuo & Nafe (1962)	Sinusoidal contact	Pertubation of boundary conditions
Herrera (1964)	Non-parallel layers	Pertubation of GF
Drake (1972)	Continental boundary	FE
Malischewski (1974)	Vertical curved interface	Alsop's method
Scheidl & Ziegler (1977)	Rigid cylindrical inclusion	Fourier series
Uberall (1977)	Buried elastic cylinder or sphere	Complex poles
Lutikov (1979)	Oblique interface	GF
Its & Yanovskaya (1983)	Subsurface curved profile	GF
Ohtsuki & Yamahara (1984)	Valley edge	FE/FD
Dravinski & Mossessian (1987)	Dipping layers; arbitrary shape	BE
Eshraghi & Dravinski (1989b)	Dipping layers; arbitrary shape	BE
Yanovskaya (1989)	Dipping layers and curved interfaces	GF
Li & Achenbach (1991)	Vertical interface zone between two materials	BE
Eshraghi & Dravinski (1991)	3-D dipping layers	BE

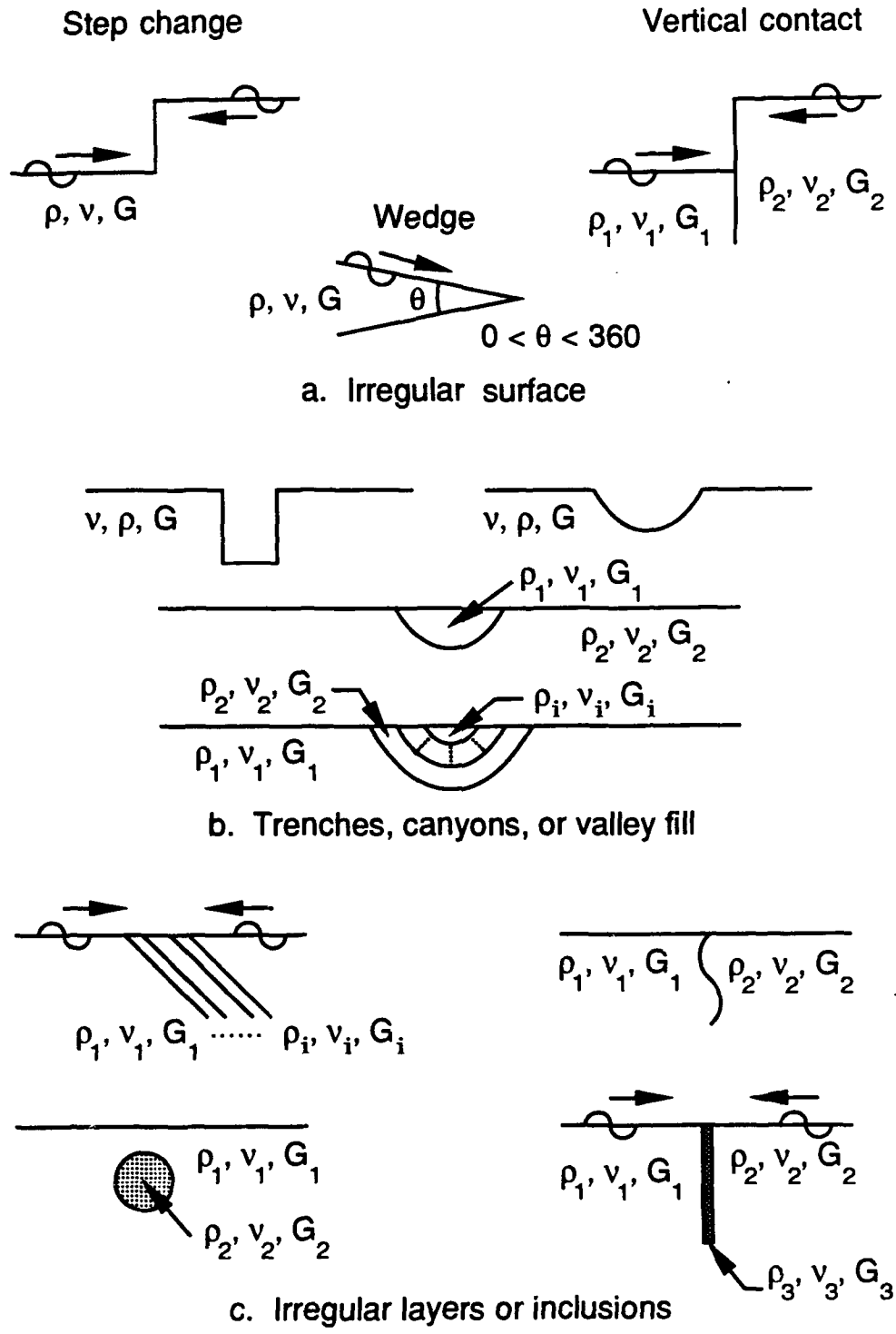


Figure 8. Schematic drawings for classification of system geometry

PART III: MATHEMATICAL FORMULATION AND COMPUTER CODE

Introduction

30. The mathematical formulation is based on simple principles of Elastodynamics, superposition, Fourier series expansion, and numerical discretization and solution procedures using the finite element method. A thorough description of the formulation, including computer implementation, is presented for completeness in this part of the report and is supplemented with derivations in Appendix B. The set of assumptions is intended to be small, to broaden the class of problems that can be solved. The primary assumption required for the condensation method described herein is that the geometry of the system and material properties are planar (do not vary in some horizontal direction). A number of other assumptions were used to derive the first generation computer code:

a. Media are isotropic,

b. Hysteretic behavior is represented by complex moduli relation,

c. Source produces vertical, steady-state excitation at one frequency,

d. Base is rigid, and

e. Distribution of loads is symmetric about y-axis.

These assumptions are not necessary and some will be phased out in future versions of the code. In addition, the computer code does not allow for transmitting boundaries in the 2-D analysis plane. Rather, the domain must be discretized to include enough area for the motions to attenuate sufficiently before being reflected back to the area of interest.

Field Equations

31. Two primary sets of variables adequately describe the effect of forces acting on linear systems -- stresses and displacements. These variables exist in the following field equations: stress equilibrium, strain-displacement, and constitutive equations. These three sets of equations are combined in terms of displacements to derive the governing equations for the problem. Wave propagation involves the effects of inertia and deformation of the media. The effects of inertia result from masses being accelerated. The derivations below apply to isotropic materials.

Stress equilibrium equations

32. The summation of stresses acting on small rectangular parallelepiped in three-dimensional Cartesian space $\mathbf{x} = x(x, y, z)$ and Newton's second law of motion neglecting body forces are used to derive the stress equilibrium equations. The equations of motion using the soil mechanics convention of compressive forces as positive and accounting for the symmetry of the Cauchy stress tensor are:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = -\rho \ddot{u} \tag{1}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = -\rho \ddot{v} \tag{2}$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = -\rho \ddot{w} \tag{3}$$

where

σ = stress components [F/L^2]

ρ = mass density [$F\text{-}s^2/L$]

$\ddot{}$ = $\partial^2 / \partial t^2$ [$1/s^2$]

These equations can be written in a much more compact form using indicial notation as:

$$\sigma_{ij,j} = -\rho \ddot{u}_i \tag{4}$$

Strain-displacement equations

33. The strain-displacement equations (in some technical fields referred to as compatibility equations) are derived from small strain theory. The equations for a displacement field $\mathbf{u} = u(u, v, w)$ are:

$$\varepsilon_{11} = \frac{\partial u}{\partial x} \tag{5}$$

$$\varepsilon_{22} = \frac{\partial v}{\partial y} \tag{6}$$

$$\varepsilon_{33} = \frac{\partial w}{\partial z} \tag{7}$$

$$\varepsilon_{12} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{8}$$

$$\varepsilon_{23} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{9}$$

$$\varepsilon_{13} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{10}$$

and are often referred to as engineering measures of strain. These equations can be written in more compact form using indicial notation as:

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{11}$$

Constitutive equations

34. The constitutive equations provide the means to relate stress and strain; they define the deformability of the material. Individual material layers are assumed to be homogeneous, isotropic, and visco-elastic. To begin the formulation of constitutive relations, consider the simplest case of linear elasticity proposed by Hooke. For homogeneous and isotropic conditions, there are two independent material constants λ and G (Lame's constants):

$$\sigma_{xx} = \lambda e + 2G\,\varepsilon_{xx} \tag{12}$$

$$\sigma_{yy} = \lambda e + 2G\,\varepsilon_{yy} \tag{13}$$

$$\sigma_{zz} = \lambda e + 2G\,\varepsilon_{zz} \tag{14}$$

$$\tau_{xy} = 2\,G\,\varepsilon_{xy} \tag{15}$$

$$\tau_{yz} = 2\,G\,\varepsilon_{yz} \tag{16}$$

$$\tau_{xz} = 2\,G\,\varepsilon_{xz} \tag{17}$$

where

G = shear modulus
$e = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}$

Using indicial notation, these six equations reduce to:

$$\sigma_{ij} = \lambda \varepsilon_{kk}\,\delta_{ij} + 2G\,\varepsilon_{ij} \tag{18}$$

where

ε_{kk} is the indicial equivalent of e defined above

δ_{ij} is the Kronecker delta: $\begin{cases} \delta_{ij}=0 \text{ if } i \neq j \\ \delta_{ij}=1 \text{ if } i=j \end{cases}$

The constitutive relations for linear-elastic materials may also be written as:

$$\sigma_{ij} = \left(\frac{2\nu G}{1-2\nu}\right) \varepsilon_{kk}\, \delta_{ij} + 2G\, \varepsilon_{ij} \tag{19}$$

where

ν = Poisson's ratio

35. Soil is an inelastic material -- energy dissipates from friction as waves travel through it. This phenomenon is called material damping and mathematical models are used to approximate it in governing equations. One form of damping, called hysteretic, is frequency independent. Clough and Penzien (1975) describe it as a (damping) force in phase with the velocity but proportional to the displacements. This form of damping can be introduced into the formulation for frequency-domain analyses through the Correspondence Principle (Wolf 1985). This principle states that the elastic stiffness (in this case shear modulus) is replaced by a complex stiffness to obtain the damped solution. The following relationship is commonly used to model linear-hysteretic behavior for small shear strains (and small values of damping):

$$G^* = G(1 + 2i\beta) \tag{20}$$

where

G^* is complex shear modulus
β is the damping ratio [-]
$i = \sqrt{-1}$

For large shear strains and values of damping, a better approximation proposed by Udaka and Lysmer (1973) is:

$$G^* = G\,(1 - 2\beta^2 + 2i\beta\sqrt{1-\beta^2}) \tag{21}$$

The results of this study are expected to be applied at distances greater than one wavelength from the source (e.g., Nazarian and Stokoe, 1985a; Kang, 1990) where shear strains are small for synthetic sources. Therefore, Equation 20

was used. The magnitude of damping is considered to be independent of strain (Hardin and Drnevich 1972; Johnston, Toksöz, and Timur 1979; and Toksöz, Johnston, Timur 1979) for the levels of shear strains expected.

Equations of Equilibrium

36. The three sets of field equations are combined to obtain the governing equations. A stiffness formulation was chosen, that is, a relation in terms of displacemer. s (also referred to as displacement approach). These equations are associated with Navier and can be derived by substituting the strain-displacement equations into the constitutive equations, then, substituting the resulting equations into the equilibrium equations. Assuming that the body forces are zero and applying Newton's second law, the result is:

$$G^* \left(\frac{1}{1-2\nu}\right)\frac{\partial \theta}{\partial x} + G^* \nabla^2 u = -\rho \ddot{u} \tag{22}$$

$$G^* \left(\frac{1}{1-2\nu}\right)\frac{\partial \theta}{\partial y} + G^* \nabla^2 v = -\rho \ddot{v} \tag{23}$$

$$G^* \left(\frac{1}{1-2\nu}\right)\frac{\partial \theta}{\partial z} + G^* \nabla^2 w = -\rho \ddot{w} \tag{24}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

Using indicial notation:

$$G^* \left[\left(\frac{1}{1-2\nu}\right) u_{j,ji} + u_{i,jj}\right] = -\rho \ddot{u}_i \tag{25}$$

These are the partial differential equations that govern wave propagation in three-dimensional Cartesian space for homogeneous, isotropic materials with no body forces. The partial differential equation is classified as hyperbolic leading to an initial value problem.

Finite Element Method in Three-Dimensional Cartesian Space

37. The finite element method is a numerical analysis technique used to approximate the response of a continuous body by dividing the domain of interest into a discrete number of subdomains. Boundary conditions and external forces are imposed at discrete nodes where the displacements are calculated. Results can be interpolated at any point in the body through the use of interpolation functions. In general, as the subdomains become smaller, the solution converges to that of the continuum. Many textbooks describing the finite element method are available with different sets of notation. The notation used below most closely follows that used by Zienkiewicz and Taylor (1989) and Bathe (1982) although some minor additions and modifications have been made.

38. There are two basic approaches to formulating a problem using the finite element method: the (direct) displacement method and the variational method. The displacement method is the most popular and most easily understood procedure (Zienkiewicz and Taylor 1989) and was selected for this study. The displacement method can be easily used with Fourier superposition analysis in the frequency domain for the solution of elastodynamic problems.

Displacement method

39. Displacements are specified as the unknowns for the displacement method. Letting u represent the vector of displacements (u, v, w) at any point (x, y, z) and U the vector of displacements at the nodes of a finite element:

$$\mathbf{u} = \mathbf{N}\,\mathbf{U} \tag{26}$$

where N is the matrix of interpolation functions. The strains at any point can be represented as:

$$\boldsymbol{\varepsilon} = \mathbf{E}\,\mathbf{u} \tag{27}$$

where $\boldsymbol{\varepsilon}$ is a vector with six strain components:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} \tag{28}$$

and

$$\mathbf{E} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \tag{29}$$

Then,

$$\boldsymbol{\varepsilon} = \mathbf{E}\,\mathbf{u} = \mathbf{E}\,\mathbf{N}\,\mathbf{U} = \mathbf{B}\,\mathbf{U} \tag{30}$$

where B is a matrix containing the corresponding derivatives of the interpolation functions.

40. The Correspondence Principal allows the constitutive model to represent hysteretic behavior using complex moduli for solutions in the frequency domain. Superposition is valid because of this linear representation. A frequency domain solution implies that the excitation function must be periodic. Calling D the complex constitutive matrix of the material:

$$\mathbf{D} = \frac{2G^*}{1-2\nu} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{31}$$

The stress vector at any point is:

$$\sigma = \mathbf{D}\,\varepsilon \tag{32}$$

with:

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{Bmatrix} \tag{33}$$

41. Applying the principle of virtual work (for derivation refer to Appendix B) and making use of the above relations, the equations of motion become:

$$\mathbf{M}\,\ddot{\mathbf{U}} + \mathbf{K}\,\mathbf{U} = \mathbf{P} \tag{34}$$

where $\mathbf{M}$ is the mass density matrix defined by:

$$\mathbf{M} = \int_V \rho\, \mathbf{N}^T \mathbf{N}\, dv \tag{35}$$

where

ρ = mass density

and $\mathbf{K}$ is the (static) stiffness matrix defined by:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{D}\, \mathbf{B}\, dv \tag{36}$$

42. The relationships for nodal acceleration, $\ddot{\mathbf{U}}$, and displacement, $\mathbf{U}$, are derived by imposing the steady state condition. First considering the load vector:

$$\mathbf{P} = \overline{\mathbf{P}}\, e^{i\omega t} \tag{37}$$

where

$\overline{\mathbf{P}}$ = vector of amplitudes of nodal forces
ω = frequency of excitation (rads/sec)

Then the displacement vector, $\mathbf{U}$, can be written as:

$$\mathbf{U} = \overline{\mathbf{U}}\, e^{i\omega t} \tag{38}$$

where

$\overline{\mathbf{U}}$ = vector of amplitudes of nodal displacements

and the velocity and acceleration vectors are:

$$\dot{\mathbf{U}} = i\, \omega\, \overline{\mathbf{U}}\, e^{i\omega t} \tag{39}$$

$$\ddot{\mathbf{U}} = -\omega^2\, \overline{\mathbf{U}}\, e^{i\omega t} \tag{40}$$

By substituting Equations 38 and 40 into Equation 34 and canceling the exponential term, the equations of motion are:

$$(\mathbf{K} - \omega^2\, \mathbf{M})\, \overline{\mathbf{U}} = \overline{\mathbf{S}}\, \overline{\mathbf{U}} = \overline{\mathbf{P}} \tag{41}$$

where $\overline{\mathbf{S}}$ is the dynamic stiffness matrix of the system defined by:

$$\overline{\mathbf{S}} = \mathbf{K} - \omega^2\, \mathbf{M} \tag{42}$$

The dynamic stiffness matrix is complex and a function of frequency. Equation 41 can be solved using matrix operations incorporated in various solution algorithms ("solvers").

43. The formulation to this point is specific to steady-state, frequency-domain analyses for homogeneous and isotropic materials. The formulation is applicable to analyses in one, two, and three dimensions and any element configuration. Henceforth, the formulation will be specific to the remaining assumptions and requirements of this study.

3-D finite element

44. A three-dimensional, isoparametric, finite element with 16 nodes was chosen to implement the condensation formulation described in the next section. A schematic of this finite element are shown in Figure 9. Each node has three degrees of freedom. The element has 8 nodes on the two x-z planes -- one at each corner and one at the mid-points on the edges -- and six nodes

on the x-y and y-z planes. Therefore, quadratic interpolation exists in the x- and z-directions (ξ and ζ in isoparametric space) and linear interpolation exists in the y-direction (η). One of the variables in the formulation is the discretization distance in the y-direction for the Fourier expansion, Δy , which can be varied without need to rediscretize the geosystem.

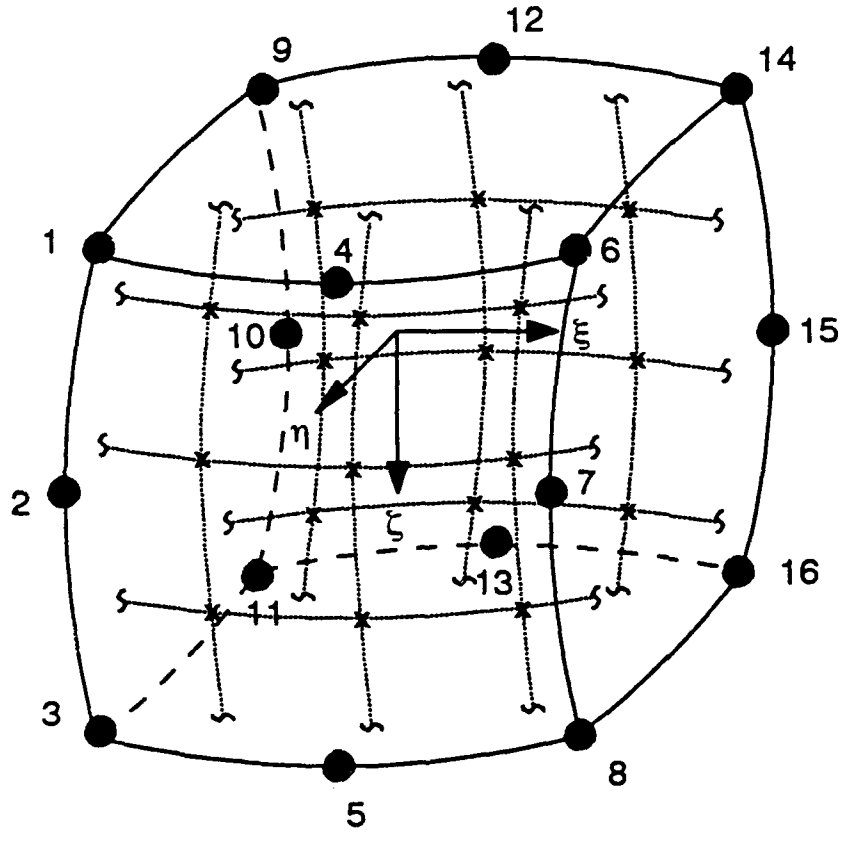


Figure 9. Specialized 16-node isoparametric finite element

45. Equations 35 and 36 can now be stated in more specific terms using the transformed space:

$$\mathbf{M} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{N}^T \mathbf{N} \, d\xi d\eta d\zeta \tag{43}$$

$$\mathbf{K} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T \mathbf{D} \mathbf{B} \, |\mathbf{J}| \, d\xi d\eta d\zeta \tag{44}$$

where | J | is the determinant of the Jacobian matrix for the 3-D finite element.

Fourier superposition

46. Fourier superposition is a three-step solution process for linear systems that involves a forward transformation into a wavenumber domain, the calculation of a solution to Equation 41 at a number of increments, and the determination of the total solution through an inverse transformation of all incremental solutions. A time-temporal frequency transform pair of a load function p are:

$$p(\omega) = \int_{-\infty}^{+\infty} p(t)\ e^{-i\omega t}\ dt \tag{45}$$

$$p(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} p(\omega)\ e^{i\omega t}\ d\omega \tag{46}$$

Similarly, the distance-spatial frequency (wavenumber) transform pair for expansion in the y-direction are:

$$p(m) = \int_{-\infty}^{+\infty} p(y)\ e^{imy}\ dy \tag{47}$$

$$p(y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} p(m)\ e^{-imy}\ dm \tag{48}$$

where

m = wavenumber (spatial circular frequency) in y-direction

Fourier superposition applied in both the time and y-spatial domains leads to:

$$p(m,\omega) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(y,t)\ e^{-i(\omega t - my)}\ dtdy \tag{49}$$

$$p(y,t) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(m,\omega)\ e^{i(\omega t - my)}\ d\omega dm \tag{50}$$

The corresponding transformation equation for displacements is:

$$u(y,t) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} u(m,\omega)\ e^{i(\omega t - my)}\ d\omega dm \tag{51}$$

47. Making the load vector specific to steady-state vibrations with constant amplitude, the time-temporal frequency transform pair reduce to:

$$p(\omega) = \bar{p} \tag{52}$$

$$p(t) = \bar{p}\, e^{i\omega t} \tag{53}$$

where $\bar{p}$ is used to represent amplitude which allows Equations 49 and 50 to be reduced to:

$$p(m)_\omega = \bar{p} \int_{-\infty}^{+\infty} p(y)\ e^{imy}\, dy \tag{54}$$

$$p(y,t) = \bar{p}\, \frac{e^{i\omega t}}{2\pi} \int_{-\infty}^{+\infty} p(m)_\omega\, e^{-imy}\, dm \tag{55}$$

for a specific ω. The corresponding equations for displacements are:

$$u(m)_\omega = \bar{u} \int_{-\infty}^{+\infty} u(y)\ e^{imy}\, dy \tag{56}$$

$$u(y,t) = \bar{u}\, \frac{e^{i\omega t}}{2\pi} \int_{-\infty}^{+\infty} u(m)_\omega\, e^{-imy}\, dm \tag{57}$$

Element condensation

48. The process of element condensation is the key aspect of the reduction of computational effort. Element condensation refers to the process of reducing the number of degrees of freedom by relating points adjacent in the y-direction using the functional relationship of the Fourier expansion. The dependent degrees of freedom are then eliminated by expressing them in terms of the in-plane degrees of freedom. In this case, the degrees of freedom corresponding to the nodes outside of the x-z plane are eliminated. Each node in the two-dimensional mesh maintains three degrees-of-freedom.

49. Consider an arbitrary discretized model of a physical system that meets the requirement of uniform geometry and material properties in one

direction such as that shown in Figure 10. The coordinate system is chosen to have the z-direction positive down and the other in-plane direction to be x. Consider three vertical planes separated by a distance of Δy at some arbitrary location along the geosystem. The 3-D dynamic stiffness matrix for any element between the slices, such as that shown in Figure 11a, is calculated using Equations 43, 44, and then 42. The dynamic stiffness matrix for a single element can be partitioned as:

$$\bar{\mathbf{S}} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} \\ \bar{S}_{21} & \bar{S}_{22} \end{bmatrix} \tag{58}$$

where the subscripts "1" and "2" refer to the degrees of freedom on the positive and negative face in the y-direction, respectively. The assemblage of the dynamic equations for any two finite elements adjacent in the y-direction, as shown in Figure 11b, can be reduced by canceling the time-dependent exponential term on each side to:

$$\begin{bmatrix} \bar{S}^{+}_{11} & \bar{S}^{+}_{12} & 0 \\ \bar{S}^{+}_{21} & \bar{S}^{+}_{22}+\bar{S}^{-}_{11} & \bar{S}^{-}_{12} \\ 0 & \bar{S}^{-}_{21} & \bar{S}^{-}_{22} \end{bmatrix} \begin{Bmatrix} \bar{U}_b \\ \bar{U}_a \\ \bar{U}_c \end{Bmatrix} = \begin{Bmatrix} \bar{P}_b \\ \bar{P}_a \\ \bar{P}_c \end{Bmatrix} \tag{59}$$

where

"+" denotes element in positive y-direction (from Δy to 0)
"-" denotes element in negative y-direction (from 0 to $-\Delta y$)
"a" denotes the degrees of freedom on face a (i.e., at y= 0)
"b" denotes the degrees of freedom on face b (i.e., at y= $+\Delta y$)
"c" denotes the degrees of freedom on face c (i.e., at y= $-\Delta y$)

50. Using the Fourier expansion described earlier (Equations 54 and 56), forces and displacements are expressed as:

$$\hat{\mathbf{P}}(m) = \int_{-\infty}^{\infty} \bar{\mathbf{P}}(y)\, e^{imy}\, dy \tag{60}$$

$$\hat{\mathbf{U}}(m) = \int_{-\infty}^{\infty} \bar{\mathbf{U}}(y)\, e^{imy}\, dy \tag{61}$$

where $\hat{\mathbf{P}}$ and $\hat{\mathbf{U}}$ are used to represent vectors of nodal forces and displacements, respectively, in m space. Rewriting Equation 59 to incorporate the Fourier expansion of loads:

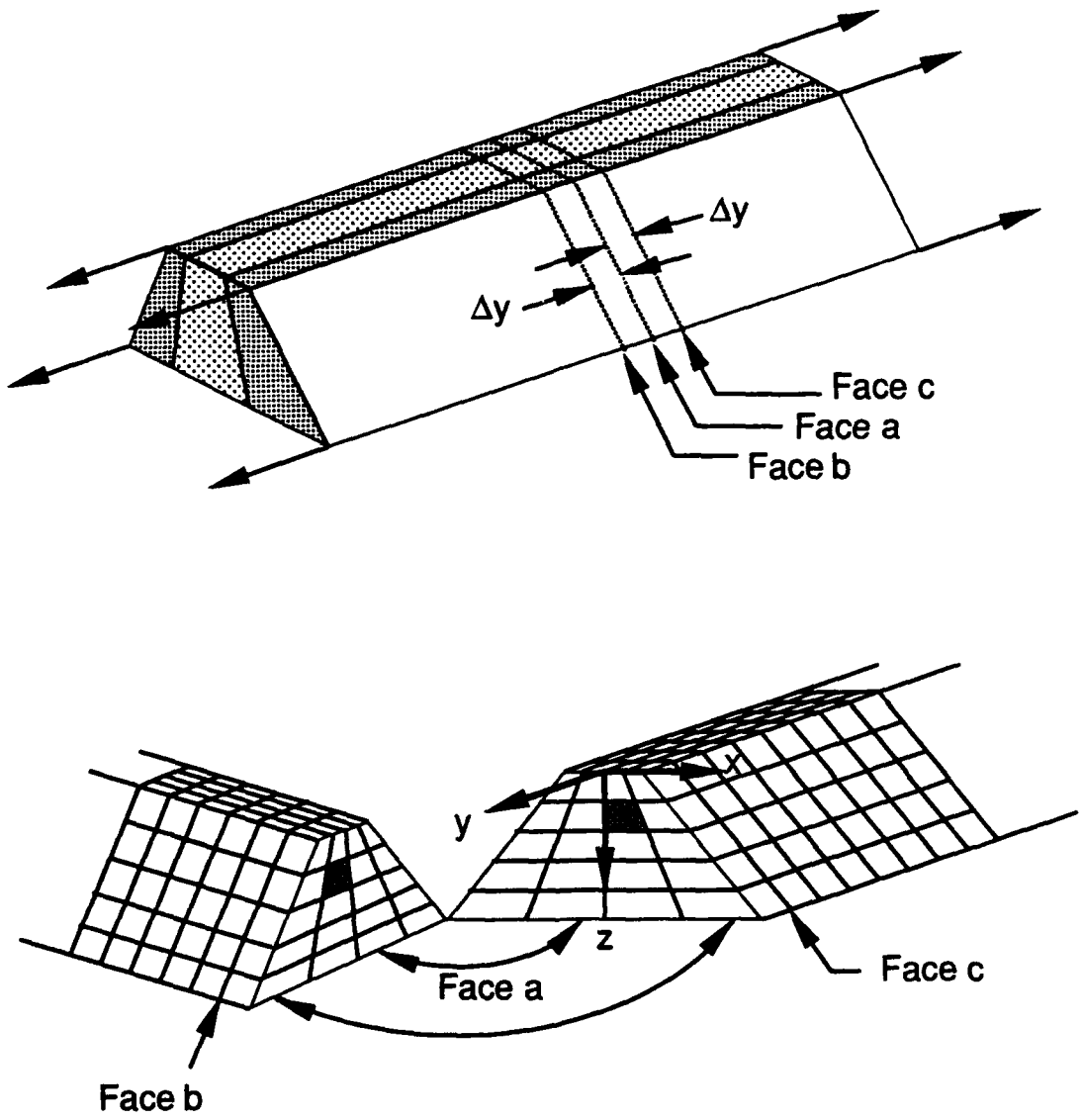


Figure 10. Example of a physical system for purposes of extracting a slice of finite elements

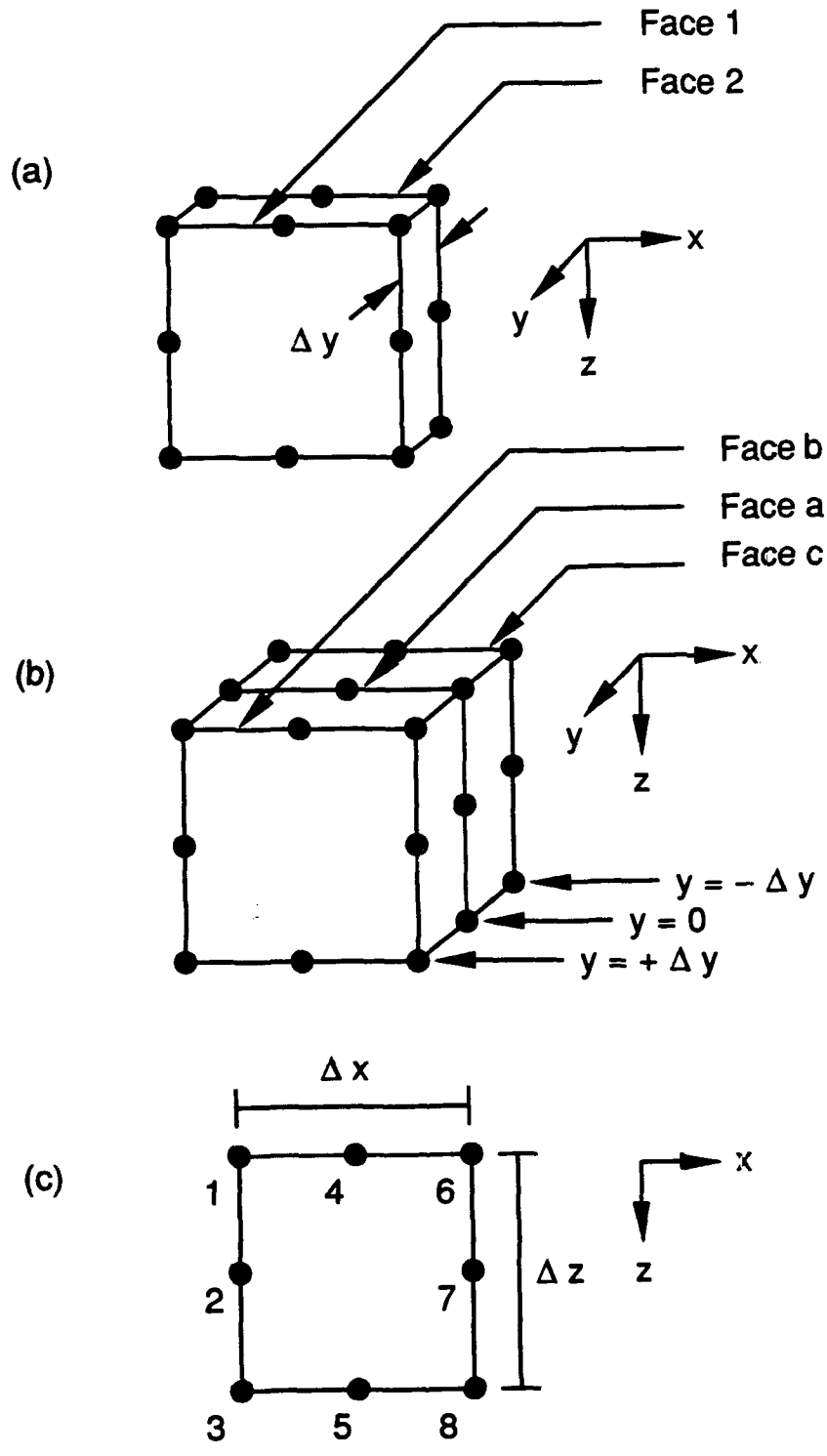


Figure 11. Condensation of finite elements adjacent in out-of-plane (y) direction

$$\begin{bmatrix} \overline{S}_{11}^{+} & \overline{S}_{12}^{+} & 0 \\ \overline{S}_{21}^{+} & \overline{S}_{22}^{+}+\overline{S}_{11}^{-} & \overline{S}_{12}^{-} \\ 0 & \overline{S}_{21}^{-} & \overline{S}_{22}^{-} \end{bmatrix} \begin{Bmatrix} \hat{U}_b(m) \\ \hat{U}_a(m) \\ \hat{U}_c(m) \end{Bmatrix} = \begin{Bmatrix} \hat{P}_b(m) \\ \hat{P}_a(m) \\ \hat{P}_c(m) \end{Bmatrix} \tag{62}$$

In the transform (m) space, the displacements on the "b" and "c" faces are related to the displacements on the "a" face at any instant in time by the simple relationships:

$$\hat{U}_b(m) = \overline{U}_a(m)\, e^{-imA_y} \tag{63}$$

$$\hat{U}_c(m) = \overline{U}_a(m)\, e^{+imA_y} \tag{64}$$

Defining:

$$\$(m) = \overline{S}_{21}\, e^{-imA_y} + (\overline{S}_{11} + \overline{S}_{22}) + \overline{S}_{12}\, e^{imA_y} \tag{65}$$

Equations 63 and 64 can be substituted into Equation 62 to get the system of equations for the equivalent two-dimensional system shown in Figure 11c:

$$\$(m)\, \hat{U}_a(m) = \hat{P}_a(m) \tag{66}$$

This formulation, then, allows the three-dimensional element with a two-dimensional geometry to be represented with an equivalent two-dimensional element. The representation of surface loads are described below.

Surface loads

51. This study focuses on the preparation for analysis of waves propagating from a synthetic, 3-D source. Vibroseis trucks generally use a rectangular platen with plan dimensions on the order of 1 by 2 m (3 by 7 ft). At large distances from the source and with large wavelengths, this area approaches a point source. Therefore, the horizontal distributions of the load considered for this study were a point load and a rectangular load of various sizes. A point source is not a physical reality and is difficult to replicate with finite elements. Kang (1990) used a point load and circular load as these were appropriate vibration source for pavement systems.

52. The formulation for equivalent nodal forces in the x-direction for point and rectangular loads are described below. The application of a point source in the finite element method is trivial. The formulation of equivalent nodal forces for rectangular loads involves integration of the force distribution in light of the interpolation function:

$$\bar{\mathbf{P}} = \int_x \mathbf{N}^T \bar{p}\, dx \tag{67}$$

For the endpoints of the 8-noded, 2-D element where z = 0 (nodes 1 and 6), this reduces to:

$$P_1 = \frac{1}{2}\left(\frac{1}{3}\xi^3 - \frac{1}{2}\xi^2\right)\Bigg|_{\xi_l}^{\xi_r} \cdot \left(\frac{\bar{p}\,\Delta x}{2}\right) \tag{68}$$

where the subscript for P refers to the node number and:

ξ_r = right-most extent of load in isoparametric space
ξ_l = left-most extent of load in isoparametric space

and

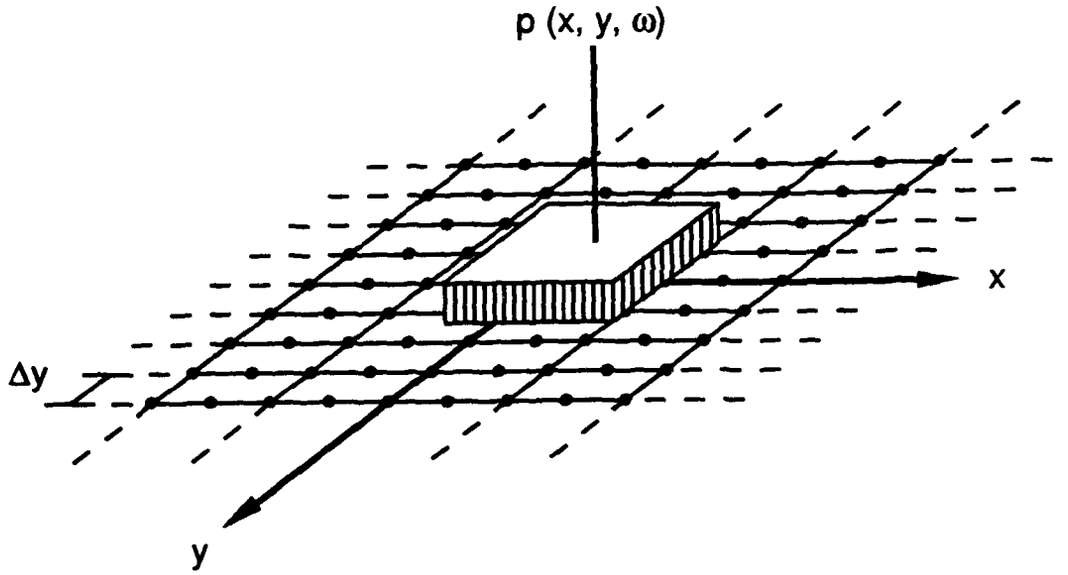
$$P_6 = \frac{1}{2}\left(\frac{1}{3}\xi^3 + \frac{1}{2}\xi^2\right)\Bigg|_{\xi_l}^{\xi_r} \cdot \left(\frac{\bar{p}\,\Delta x}{2}\right) \tag{69}$$

For the midpoints of the 8-noded element, this reduces to:

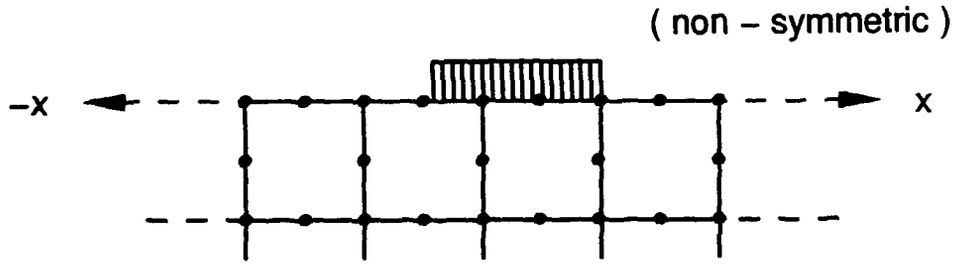
$$P_4 = \frac{1}{2}\left(\xi - \frac{1}{3}\xi^3\right)\Bigg|_{\xi_l}^{\xi_r} \cdot \left(\frac{\bar{p}\,\Delta x}{2}\right) \tag{70}$$

The distribution of forces applied to the platen is assumed to be uniform and therefore the integration reduces to simple algebra. For example, a continuous, uniform load with a total magnitude of unity ($p \cdot \Delta x = 1$), the equivalent nodal forces are 1/6 for the endpoints and 4/6 for the midpoint.

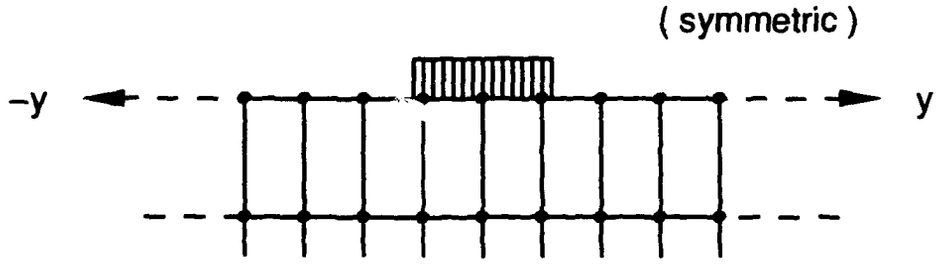
53. The process of converting loads from the time-spatial domain to the frequency-wavenumber domain is depicted in Figures 12 through 14. In Figure 12a, an arbitrary rectangular pressure is applied vertically at the ground surface. The distribution in the x- and y-directions are shown in Figures 12b and 12c, respectively. The distribution in the y-direction is



a. Oblique view

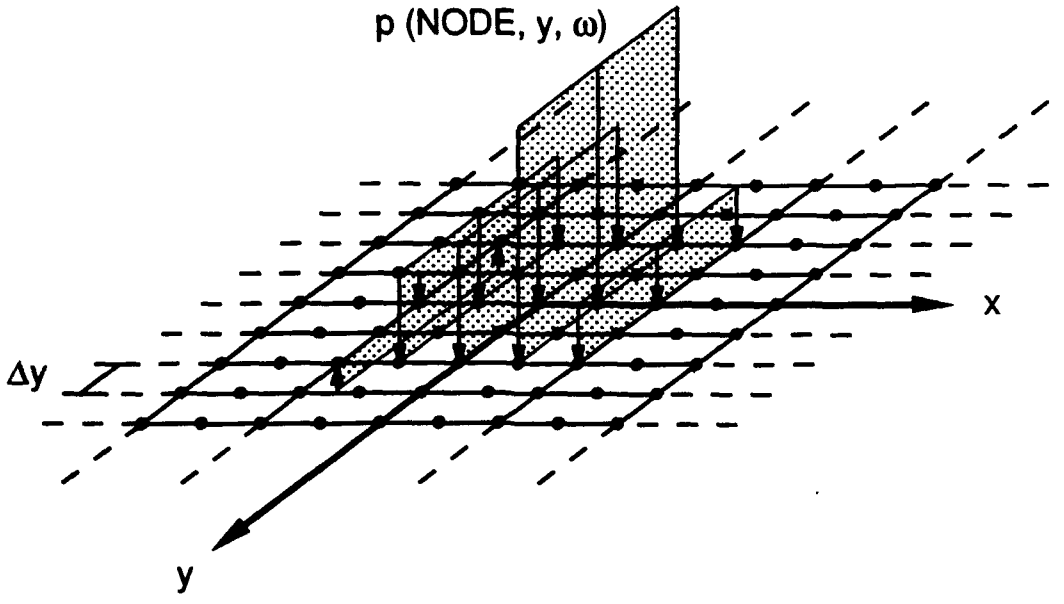


b. Profile at y = 0

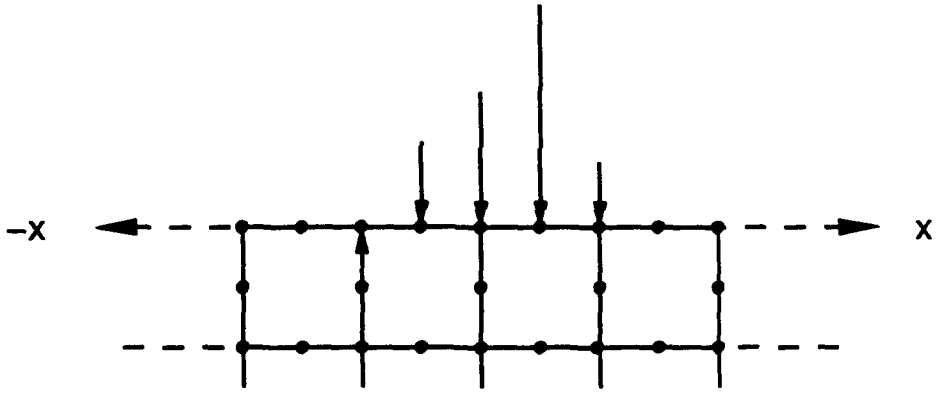


c. Profile at x = 0

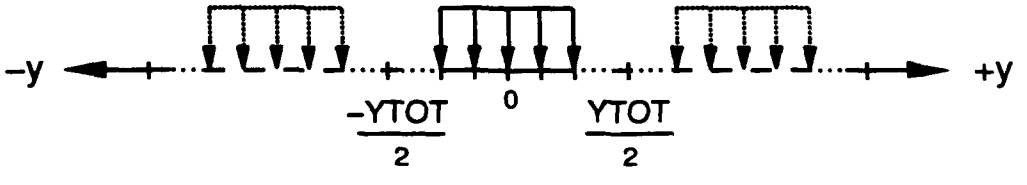
Figure 12. Distribution of arbitrary rectangular pressure



a. Oblique view

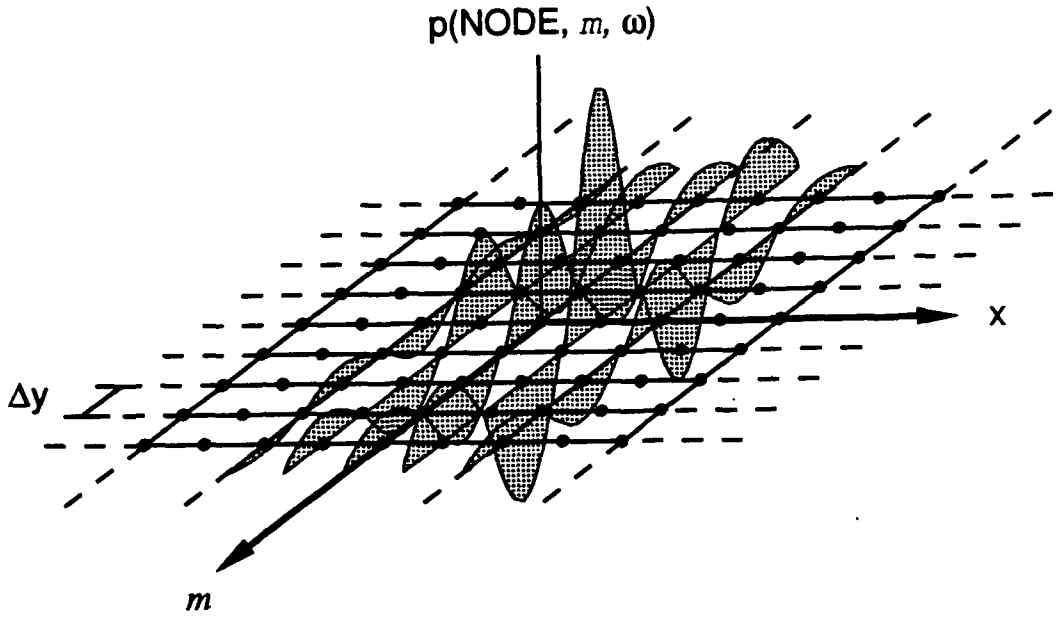


b. Profile at y = 0

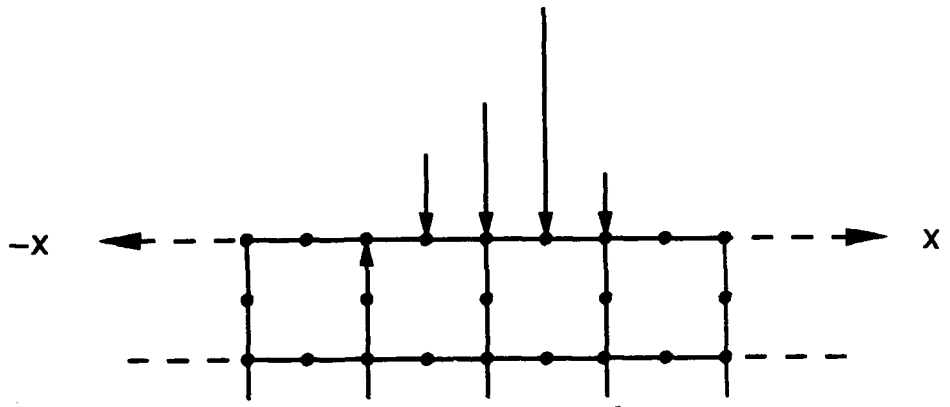


c. Profile at x = 0

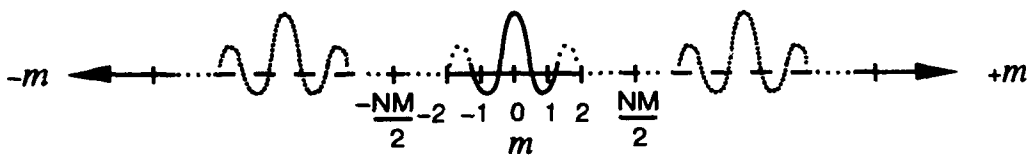
Figure 13. Distributions of nodal forces resulting from arbitrary rectangular pressure



a. Oblique view



b. Profile at y = 0



c. Profile at x = 0

Figure 14. Distributions of Fourier expansion of loads resulting from arbitrary rectangular pressure

assumed to be symmetric to reduce the number of operations by nearly a factor of two. The equivalent nodal forces in the finite elements on the x-z plane are shown in Figure 13. After the Fourier transformation in the y-direction, the distribution of forces in the x-m space is shown in Figure 14.

Time-dependent displacements

54. The real-valued, time-dependent displacements may be obtained from the calculated complex displacements, U. If the forcing function is of the form sin ωt, then:

$$u_i = A_i SIN(\omega t) + B_i COS(\omega t) \tag{71}$$

If the forcing function is of the form cos ωt, then:

$$u_i = A_i COS(\omega t) - B_i SIN(\omega t) \tag{72}$$

where

A_i = real part of complex displacement amplitude at node i

B_i = imaginary part of complex displacement amplitude at node i

For the analysis of the vibrations produced by a Vibroseis, Equation 72 is more appropriate. The phase angle of motion, ϕ , is calculated by:

$$\phi = \tan^{-1}\left(\frac{B_i}{A_i}\right) \tag{73}$$

Computer Implementation

55. The system of computer programs written for the solution of the dynamic displacements in planar geosystems includes (pre-processing) mesh generation and visualization routines, the primary finite element code, and visualization (post-processing) routines to analyze the results. Computer codes were written using the Fortran 77 and C computer languages running on the U.S. Army CRAY Y-MP supercomputer and Silicon Graphics workstations supported at WES. A listing of FORTRAN computer code vib3 is contained in Appendix C. A sample input and corresponding output file are provided in Appendices D and E, respectively. Basics of the implementation in the primary computer code, *vib3*, are presented below.

Computer code vib3

56. The function of *vib3* is to read in the parameters defining the mesh, material properties, load, and Fourier expansion parameters and solve for the corresponding complex displacements. This program evolved from an unnamed finite element code used for instructional purposes in a graduate engineering course entitled "The Finite Element Method" taught by Prof. John Tassoulas in 1988 at the University of Texas at Austin. The basic framework of variable storage in a massive single-scripted array and solution using a frontal solver were kept as well as a subroutine to modify element stiffness for boundary conditions. The remainder of the subroutines were written specifically for this study.

57. The interaction of subroutines in *vib3* is represented using the flowchart presented in Figure 15. The subroutines are called consecutively from the **MAIN** program. **MAIN** contains one DO LOOP to allow for the analysis of multiple frequencies and a DO LOOP that creates the matrices and solves for displacements at each increment of **m**. General descriptions of each of the subroutines listed are provided below.

Input of mesh and problem parameters

58. The subroutines **MAIN**, **DATAIN**, and **ELP3D** read information from the input file regarding system geometry, boundary conditions, material properties, Fourier expansion, and code operation (input/output). At first, **MAIN** is used to read a problem title and basic variables that affect the allocation of array space and define the scope of calculations. Once the key parameters have been read and the array lengths are established, the allocated program memory for the real and complex arrays are checked prior to execution. Then, subroutine **DATAIN** is used to read the 2-D nodal coordinates (the third dimension, Δy, is constant) and boundary conditions, the nodes of primary interest for analysis, and the element connectivities and material correspondence. Boundary conditions in the x- and z-directions are specified in the input file. Subroutine **ELP3D** is used to read the material properties of each of the materials designated and calculate the components of the constitutive matrix (Equation 31).

Load vector

59. Subroutines **YLOAD** and **XLOAD** are used to define the extent of the load distribution and create the load vector, $\bar{\mathbf{P}}$. The option for either a point load or a rectangular load exists. The magnitude of the load is an

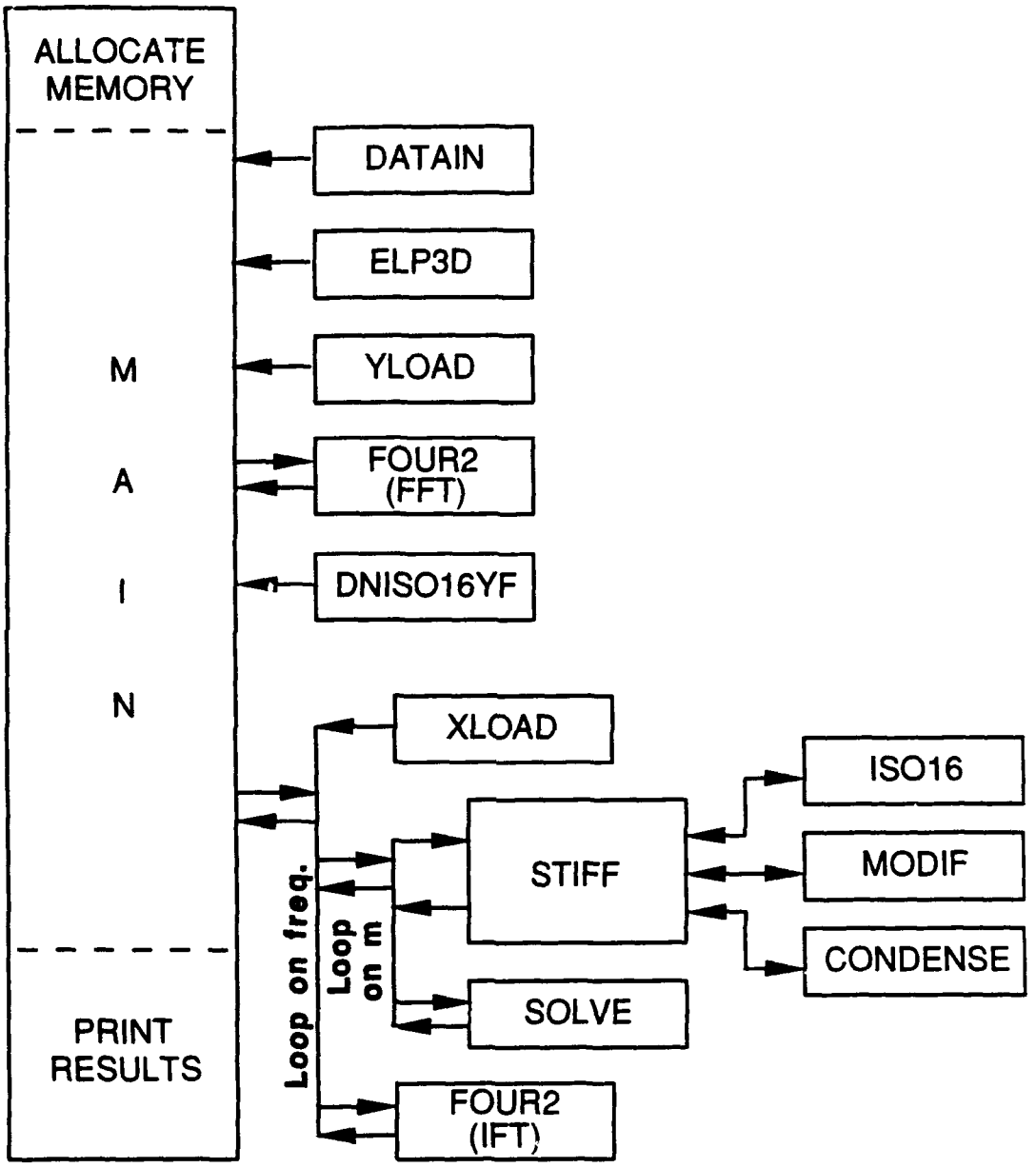


Figure 15. Flowchart of primary subroutines in *vib3*

input value and is assumed to be constant over the specified area and symmetric about the x-z plane (y = 0 axis). This assumption reduces the time to solution by a factor of two by taking the solution of displacements in the +m direction and mirroring them in the -m direction in the wavenumber space.

60. The subroutine **YLOAD** is called first because these operations are independent of most other operations and must be done before the Fourier expansion. This subroutine reads the magnitude of the load and horizontal extent from the point y = 0. The distribution is stored in terms of a discretized set -- a single value of magnitude at each Δy within the extent of the load distribution. This set is used directly by the Fourier transform algorithm described later. Next, the subroutine **XLOAD** calculates the vertical nodal forces using Equations 68 through 70. The extent of the load is specified by the smallest and largest values of x. Algorithms determine the affected elements and nodes.

61. An option to use normalized loads is available which is valuable for validation study and parametric analysis. Comparisons between results from different load geometries and configurations are aided by normalizing the load thus producing the magnitudes of displacements that are independent of the distribution of the load. Normalization refers to producing a total load (pressure times uniformly loaded area) of unity. This option is enacted by two steps:

a. Specifying a magnitude of load that equals the inverse of the width of the load in the x-direction, and

b. Using a single FORTRAN statement in subroutine **YLOAD** that divides the magnitude by the number of non-zero pre-expansion terms in the y-direction.

The statement in subroutine **YLOAD** is included in the computer code but must be switched on and off manually through the use of a FORTRAN comment statement.

Fourier expansion

62. A Fourier expansion of the distribution of load in the y-direction follows the specification of the load parameters in subroutine **YLOAD**. The inverse transformation of displacements to the spatial domain is conducted after the solution process is completed for each value of m. The 3-D load vector is created after the forward transform by multiplying the equivalent nodal forces (from **XLOAD**) and the m-independent, complex transform (from **YLOAD** and **FOUR2**). This vector is dependent on the value of m so this

operation is performed within the DO LOOP on m. The multiplication of these values creates a load distribution that can be visualized by looking at Figure 14. Nodal forces on elements outside the extent of the load are zero.

63. The Fourier transformations are performed using the Fast Fourier Transform (FFT) algorithm proposed by Cooley and Tukey (1965) in the subroutine FOUR2 and other dependent subroutines established in the 1970's at the Massachusetts Institute of Technology. The discrete values of the transform are:

$$\bar{P}(m_n) = \Delta y \sum_{j=0}^{N-1} p(y_j) e^{2\pi i \frac{n_j}{N}} \tag{74}$$

where

$$n = 0, 1, 2, \ldots, N-1$$

and:

$$p(y_j) = \frac{\Delta m}{2\pi} \sum_{n=0}^{N-1} \bar{P}(m_n) e^{-2\pi i \frac{n_j}{N}} \tag{75}$$

where

$$j = 0, 1, 2, \ldots, N-1$$

and

N = number of sampling points (power of 2)
$y_j = j \cdot \Delta y$
YTOT = $N \cdot \Delta y$
$m_n = n\Delta m$
$\Delta m = 2\pi/YTOT$

Element stiffness

64. The finite element stiffness and mass matrix are created through the use of subroutines DNISO16Y and STIFF. Subroutine STIFF calls subroutines ISO16, MODIF, and CONDENSE. Subroutine DNISO16Y is used to calculate the interpolation functions and their derivatives at each of the integration points. Since the same finite element is used throughout and these values are independent of Cartesian coordinates, this routine is called once. Subroutine STIFF collects the nodal coordinates, constitutive matrix, and other parameters for the element being considered and calls the subroutine ISO16 to perform the numerical integration and convert the static stiffness matrix into

a dynamic stiffness matrix. Subroutine **MODIF** is called by **STIFF** to modify the stiffness matrix and load vector in the case of rigid boundary conditions.

65. A numerical integration technique is used to calculate the 3-D dynamic stiffness matrix for each finite element. This technique begins with a transformation of the spatial coordinates from Cartesian space $\mathbf{x} = f\ (x, y, z)$ to a normalized coordinate system $\underline{\mathbf{x}} = f\ (\ \xi\ ,\ \eta\ ,\ \zeta\ )$ centered within the range of integration (refer to Figure 9). Integrals are replaced by summations computed at specific (integration) points and are scaled by appropriate weighting factors. Equations 43 and 44 used to calculate mass and stiffness are rewritten as:

$$\mathbf{M} = \sum_{i}^{3} \sum_{j}^{2} \sum_{k}^{3} \left(\mathbf{N}^{T} \mathbf{N}\ |\mathbf{J}| \right)_{ijk} \tag{76}$$

$$\mathbf{K} = \sum_{i}^{3} \sum_{j}^{2} \sum_{k}^{3} \left(\mathbf{B}^{T} \mathbf{D}\ \mathbf{B}\ |\mathbf{J}| \right)_{ijk} \tag{77}$$

66. Eighteen integration points were used to derive the element stiffness. The exact location of these points are defined by coordinates of 0 and $\pm\ \sqrt{3}\ /\ \sqrt{5}$ in the x- and z-directions and with $\pm\ 1\ /\ \sqrt{3}$ for the y-direction. The weighting factors are 8/9 for the midpoint, 5/9 for the endpoints, and 1 for the y-direction (Stroud and Secrest 1966).

Condensation

67. The 3-D dynamic stiffness matrix and load vector for the element are condensed to the equivalent 2-D matrix and vector, respectively, using Equation 65 as coded in subroutine **CONDENSE**. The size of the stiffness matrix changes from 48 by 48 components to 24 by 24; the load vector changes from 48 by 3 to 24 by 3. After **CONDENSE**, these element-dependent components are transferred to the solver.

Frontal solver

68. The solution algorithm used is called a frontal solver. This process involves gathering the dependent equations necessary to determine the value of a particular degree of freedom. A detailed presentation of frontal solvers for positive-definite matrices was made by Irons (1970).

69. The solver used for this study, **SOLVE**, was created and refined over several years through the efforts of Profs. C. P. Johnson and Eric B. Becker at the University of Texas at Austin. This solver can

accommodate symmetric and non-symmetric matrices although the option for non-symmetric matrices is not necessary when using the displacement-based approach with elastic materials. The UT solver was modified slightly to allow for the solution of matrices with complex-valued components.

Other subroutines

70. Other subroutines were written to perform various systematic operations that ensured conformity and print portions of stiffness matrices and load vectors. The subroutine **SYMSM** is used to check the symmetry of the stiffness matrix at any stage of the calculation and is a useful tool for recognizing and debugging errors in parameters defining the mesh. A symmetry tolerance is specified in the input file. The subroutines **PRNTRHS** and **PRNTSM** allow the load vector and stiffness matrix, respectively, for any element and "m" step to be saved.

Other options

71. Symmetric problems (in the x-direction) may be solved with the present formulation to reduce the computation time by greater than a factor of two. The boundary conditions along the line of symmetry should be fixed in the x-direction (IBC=1) and free in the z-direction. The load width is now the half-width with the same magnitude (unless a normalized load is used in which case the magnitude equals the inverse of the half-width). No other special considerations are required. Symmetric problems were considered for validation and parametric analyses as described in the next two parts of this report.

PART IV: VALIDATION STUDIES

General

72. Validation studies and parametric analyses were used to prove that the formulation and computer implementation are sound, accurate, and stable for the limited problem class to which accurate solutions are available. The findings of validation studies are not mutually exclusive from the parametric analyses because the definition of the problems for validation should conform somewhat to the findings of parametric analyses. The results of the validation studies are described below; the parametric analyses are described separately in Part V.

73. The best form of validation consists of comparing the results between a subject program and exact mathematical relationships for several different problems. Comparisons with measured data or prototype testing provide a constructive means to confirm findings when conducted under certain controlled conditions. These comparisons are not appropriate as the primary means of validation, however. Comparisons with other numerical approximations are even less appropriate for validation. Validation of *vib3* through comparisons with analytical results is possible only for the simplest class of planar geometry -- a horizontally layered system extending to infinity. Green's function solutions formulated for axi-symmetric problems by Kausel (1981) were used exclusively. Some minor differences in displacements may exist between the Green's function solutions and the 2-D approximations because the shape of the load is different -- disk loads were used for the axi-symmetric problem and square loads were used for this study. The same total area and total load of unity were used to minimize these differences. The model systems used to validate the computer code are described in the next section.

74. The validation studies described in this part pertain to variations in system geometry, material properties, and frequency of excitation. The dynamic vertical displacements are of primary interest because they predominate in surface motions caused by vertical excitations. Moreover, vertical vibrations are normally measured in non-destructive testing techniques such as the SASW method. Measurements are likely to be made both perpendicular and parallel to the structure of the system (x- and

y-directions, respectively). Therefore, the results are presented in terms of the variations of real and imaginary components of dynamic displacement in both the x-direction (calculated at nodes on the y = 0 plane) and the y-direction (calculated at node beneath the centroid of the load and expanded out in the y-direction). Most comparisons are made at the free (ground) surface although a few comparisons are also made below the surface. Distances are normalized to the wavelength of Rayleigh waves, λ , for Model 1. (Note that the definition for the Greek letter λ has changed from that used previously.) The displacements are oriented positive-down to be consistent with the convention used in the formulation and correspond to the top surface (z = 0).

Test Models and Discretization Schemes

75. Four hypothetical models were created for validation studies and are shown with unit-less dimensions in Figure 16. These models were designed to represent ideal site conditions of horizontally layered soil overlying rock and realistic material properties (considering units of ft-lb-sec) while conforming to limitations of the analytical solutions. All models have the same total height (1000 units) and are assumed to overlay a rigid material. Model 1 is the simplest system -- a homogeneous medium overlying rock. The range of material properties for this medium used in the following comparisons are shown in Figure 16. The other three models consist of four homogeneous layers overlying rock with different combinations of stiffness.

76. A domain with dimensions 1000 units high and 2500 units wide was chosen, along with the material properties and frequency of excitation, to be large enough to ignore the effects of reflections and correspond to about 3λ high by 8λ wide. Three different finite element meshes were created to represent this domain and are shown in Figure 17. The domain was discretized using 4 by 10, 8 by 20, and 16 by 40 square elements. The size of these elements corresponds to 0.8λ, 0.4λ, and 0.2λ, respectively. A plane of symmetry at the left boundary, defined by x = 0, was utilized to reduce the degrees of freedom by nearly one-half.

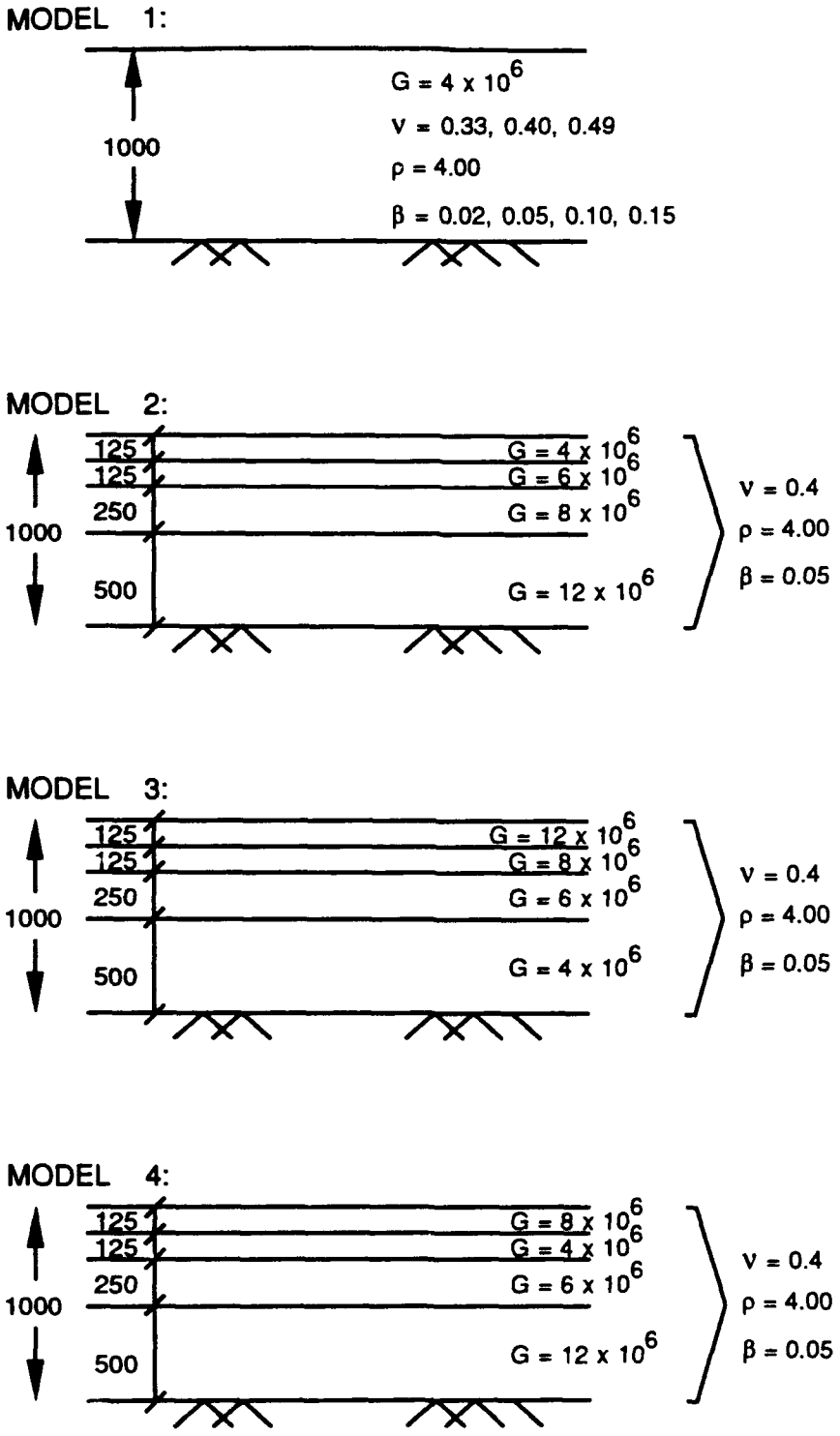
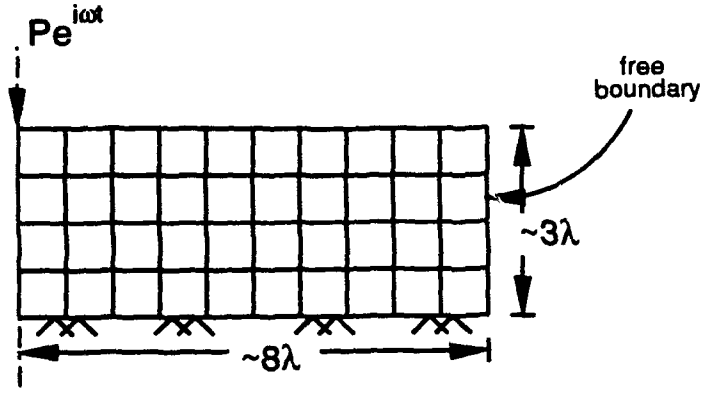
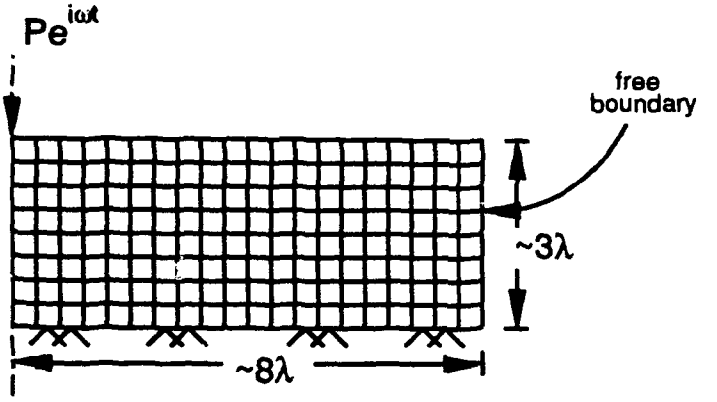


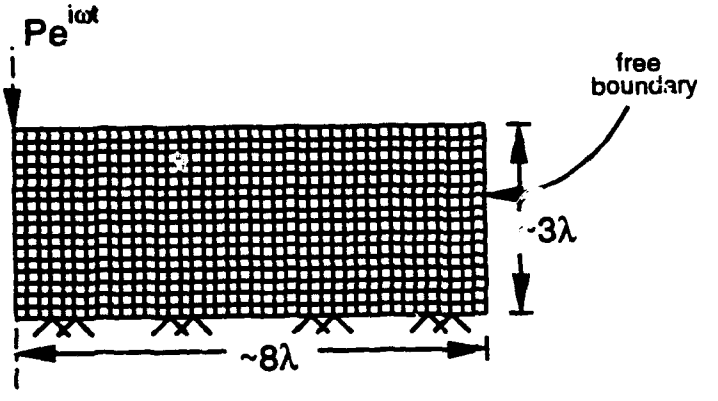
Figure 16. Test models used for validation studies



a. Coarse mesh with 4 by 10 elements



b. Median mesh with 8 by 20 elements



c. Fine mesh with 16 by 40 elements

Figure 17. Discretized models used for validation studies and parametric analyses

Analytical Solutions

77. The Green's function solutions formulated by Kausel (1981) were calculated with the computer code *PUNCH* (Kausel 1989) using a personal computer. The calculated solution approaches the exact solution as the number of layers increases. The displacements calculated using *PUNCH* correspond to a disk load with radius r and total load, P, of 1 ($= p\pi r^2$) or a point load with magnitude of unity. The only limitation of *PUNCH* that impacted the analysis is that a maximum of 30 layers can be used. The effect of this limitation was examined using Model 1 and is reported below using a frequency of excitation of 3 Hz, system damping of 2 percent, and a radius of load of 5.64 (total area of 100).

78. The effect of the number of layers on Kausel's solution for Model 1 was evaluated using four different layered systems (4, 8, 16, and 24 layers). The variation of the real and imaginary components of the complex dynamic displacement (refer to Equations 71 and 72) with (horizontal) distance (in this case radial) from the center of the load are shown in Figure 18 for these four cases. The maximum horizontal distance considered, 5λ, corresponds to about 1500 units. The results plotted in Figure 18 indicate that the number of layers can have a significant impact on the amplitude of both components. As the number of layers increases and displacements approach the true solution, the amplitudes increase. The peaks also tend to occur closer together as the true solution is approached.

79. The results have not completely converged with 24 layers, but the changes from 16 to 24 layers are small, especially at distances less than about 4λ. It appears that 24 layers is an adequate number to represent this system with the realization that the exact solution is likely to have slightly larger displacements at the peaks and possibly a more compact waveform. Twenty-five layers were used to represent Model 1 for further comparisons and validation to provide a uniform thickness of 40 units for each layer.

80. The effect of the radius of the load, r, was also examined while keeping the total load equal to unity. A point load and a distributed load with normalized radii of 0.036λ, 0.072λ, and 0.144λ (areas of 400, 1,600, and 6,400, respectively) were used and the results are shown in Figure 19. The dynamic displacements are nearly equivalent for the four cases. Perceptible

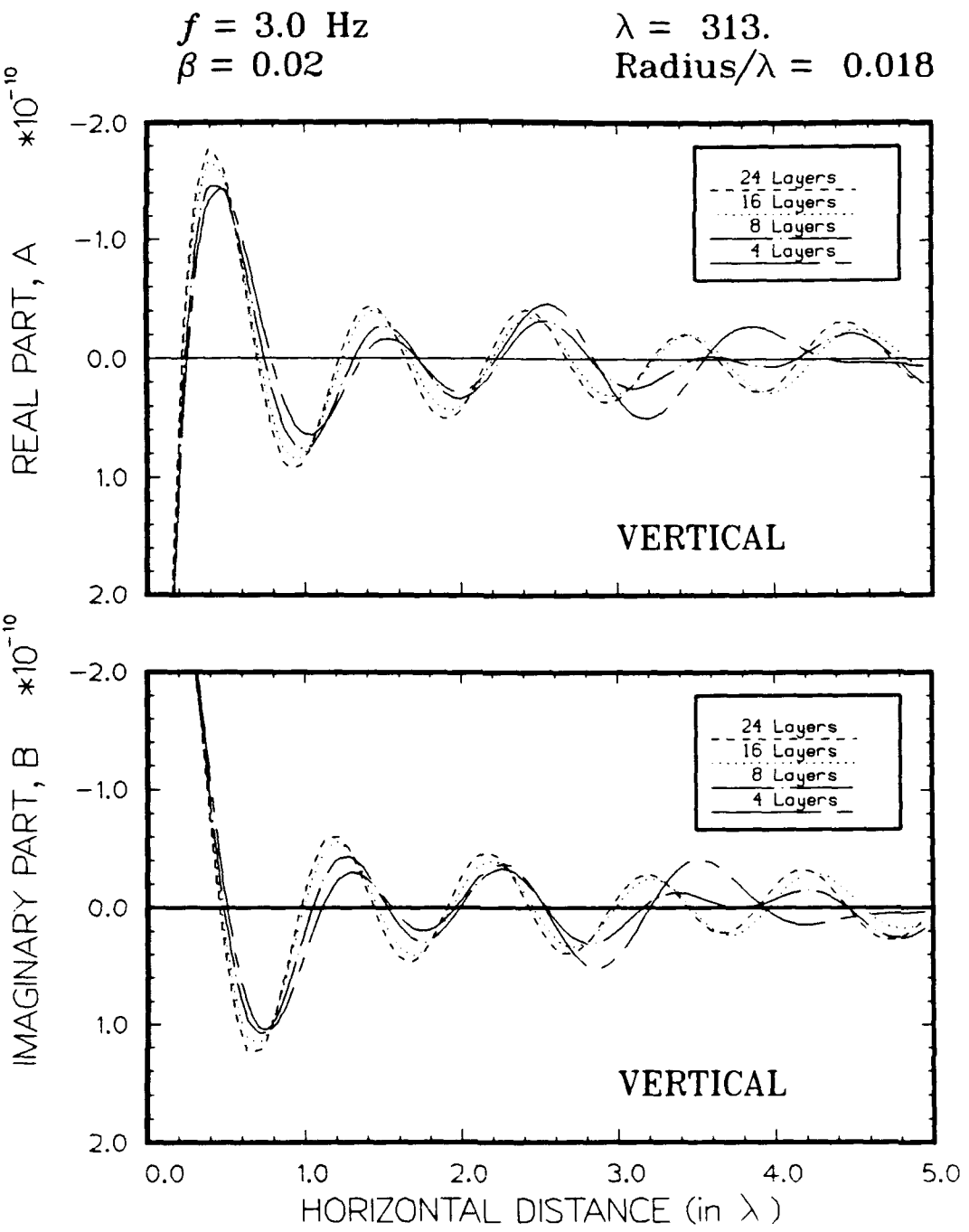


Figure 18. Comparison of dynamic displacements from Green's function solutions for various numbers of layers

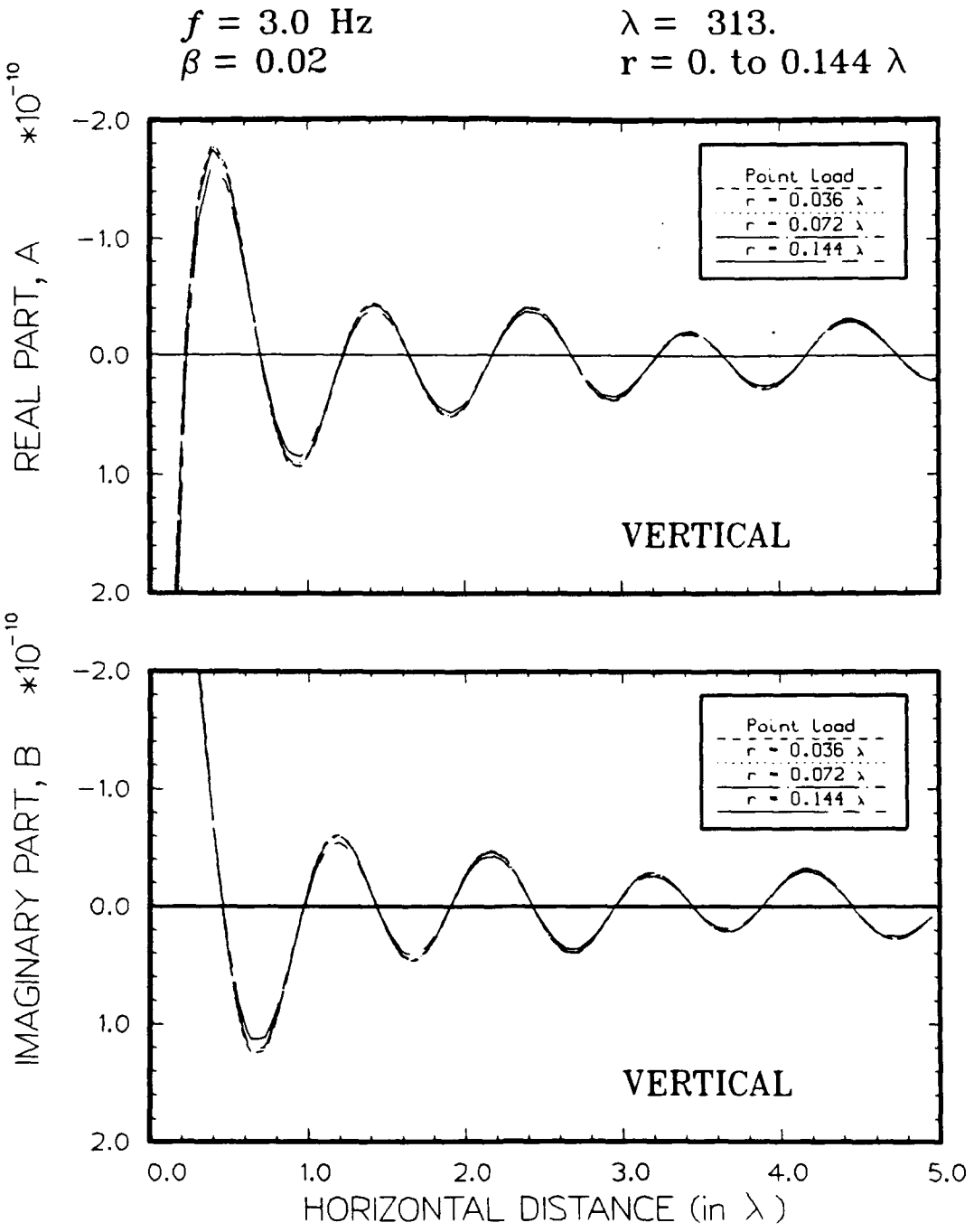


Figure 19. Comparison of dynamic displacements from Green's function solutions for various radii of loads

differences exist only for data corresponding to the largest radius, $r = 0.144\lambda$. Therefore, it is safe to conclude that the radius of the load has little effect on the dynamic displacements normalized to the total load for ratios of $r \leq 0.10\lambda$. A radius of loading equal to 0.018λ was used for the remaining comparisons which essentially represents a point load.

81. The effects of varying Poisson's ratio and the damping ratio on the dynamic vertical displacement were examined using Model 1 to facilitate some of the comparisons made later for parametric analyses. The effect of varying Poisson's ratio from 0.33 to 0.49 is shown in Figure 20. The general trend of the three relationships is that the peak values of displacement decrease and the distances between peak values increase as Poisson's ratio increases. The effect of varying damping ratio from 0.02 to 0.15 is shown in Figure 21. The general trend of the four relationships is that the peak values of displacement decrease as damping ratio increases; there is little change in the radial distance at which the peak values of real and imaginary parts occur.

82. The comparisons shown in Figures 20 and 21 bring about an important consideration for geosystems -- Poisson's ratio and damping ratio are two material properties that are difficult to determine for soils. Oftentimes, these two properties are estimated using empirical relations or data bases of measured values. Estimated values may possibly be different from true values which is a potential source of error. Errors in estimating ν and damping ratio can be significant for reasonable ranges of these properties. Variations in ν affect both the amplitude and location of peak amplitudes of the real part whereas variations in damping ratio affect primarily the amplitude.

Element Performance to Static Loads

83. The specialized 3-D finite element was evaluated for the ability to represent static response to various loads. This evaluation was accomplished by comparing the results of two approaches with analytical solutions. One approach was to place the algorithms defining the element stiffness into a static finite element computer code and examine the response of a cantilever beam. The other approach was to use *vib3* with a point load

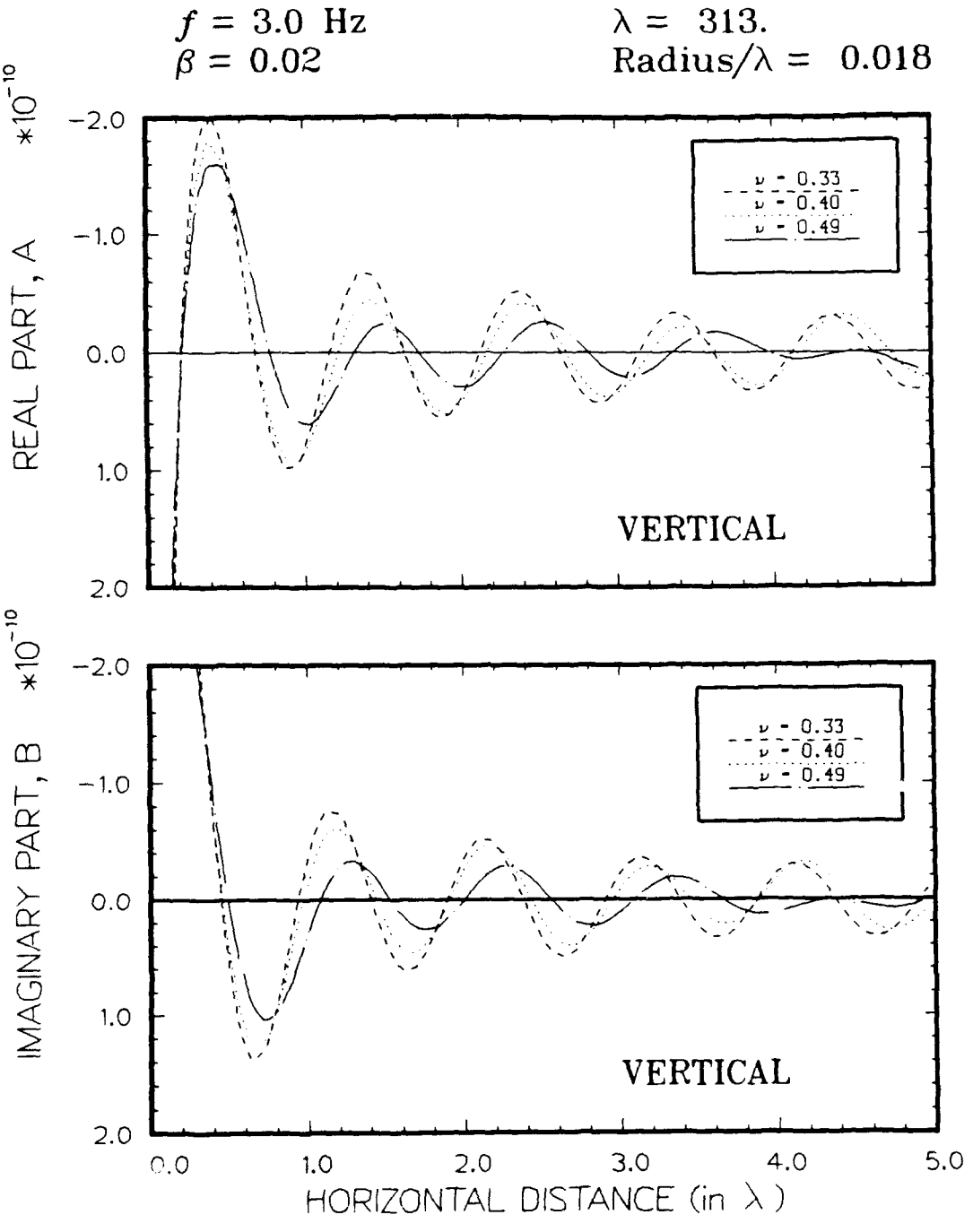


Figure 20. Comparison of dynamic displacements from Green's function solutions for various Poisson's ratios

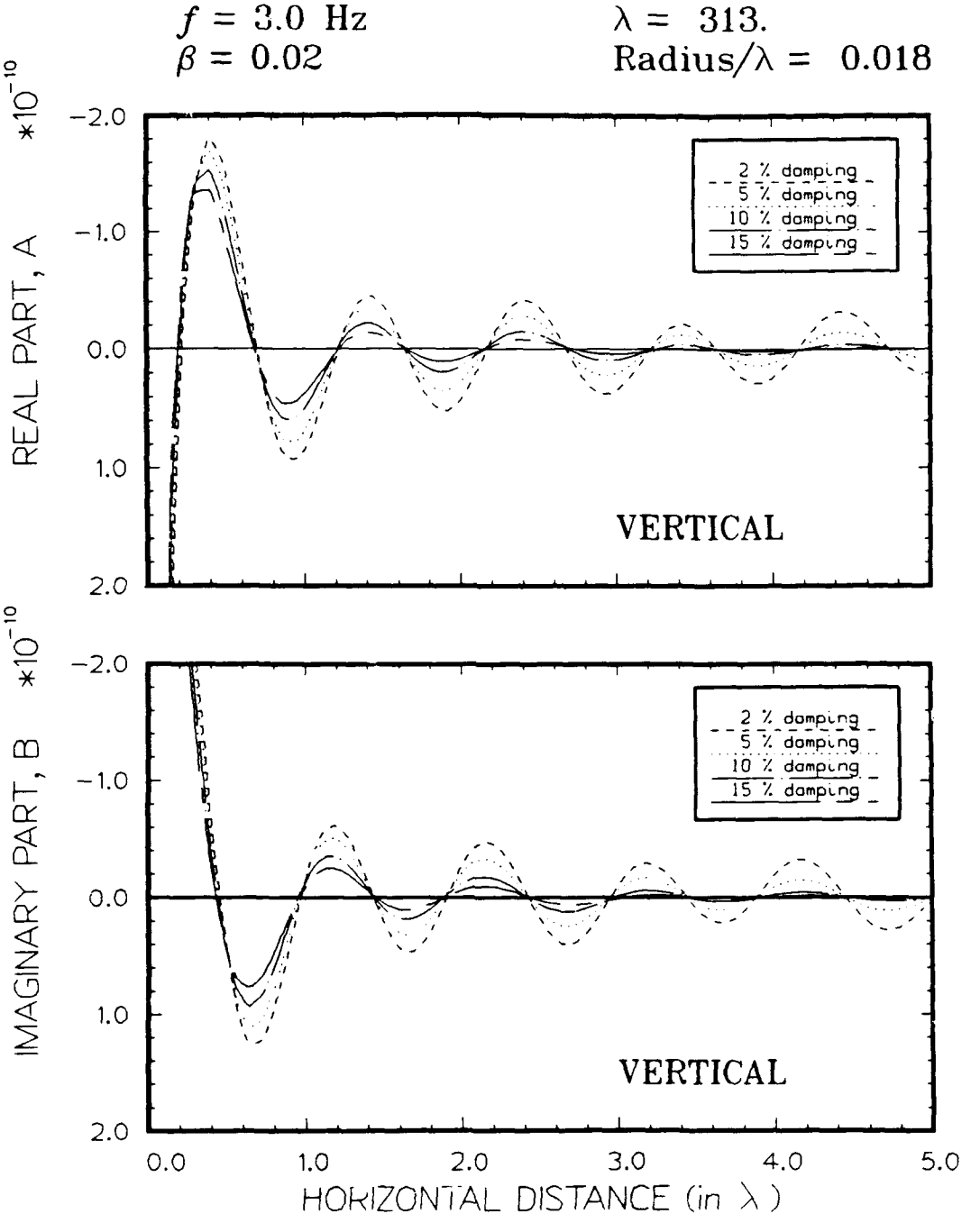


Figure 21. Comparison of dynamic displacements from Green's function solutions for various damping ratios

acting on a homogeneous body with the frequency equal to 0. Each of these are described below.

Static finite element code

84. A static finite element code was used to evaluate the specialized finite element. This program evolved from an unnamed finite element code used for instructional purposes in a graduate engineering course entitled "The Finite Element Method" taught by Prof. John Tassoulas in 1988 at the University of Texas at Austin. A cantilever beam was discretized with 2, 5, 10, 40, and 80 elements and subjected to tension, compression, and shear-induced bending loads. The effect of element shape was also evaluated by considering square, rectangular, parallelogram, and trapezoidal configurations. Comparisons between calculated and closed-form solutions for displacements and stresses were good and indicate that the algorithms defining the element stiffness are accurate for conditions of static loading.

Dynamic code

85. The static vertical displacements calculated using *vib3* with Model 1 at depths of 0, 125, 250, and 500 units (0.0λ, 0.40λ, 0.80λ, and 1.6λ, respectively) are shown in Figures 22 through 25 using the finest of the three meshes. The comparisons with Green's function solutions in the y-direction are excellent for the real part and very good for the imaginary part at distances slightly removed from the point of load (greater than 100 units). Comparisons are similar at all depths. The imaginary part should be zero at all distances but *vib3* produces non-zero values at locations close to the load. The less favorable comparisons near the point of loading are common when modeling a point load using the finite element method. These errors are normally minimized through mesh refinement near the point of loading but accuracy close to the source is not of interest for this study. The variation of vertical displacements in the x-direction calculated using *vib3* represent the exact same relationship as the variation of vertical displacements in the y-direction at all depths except $z = 0$. The real part at distances less than 150 units oscillates considerably about the other solutions.

Approximations for Dynamic Loads

86. The computer code *vib3* was used to calculate dynamic displacements for each of the four models described previously. These results

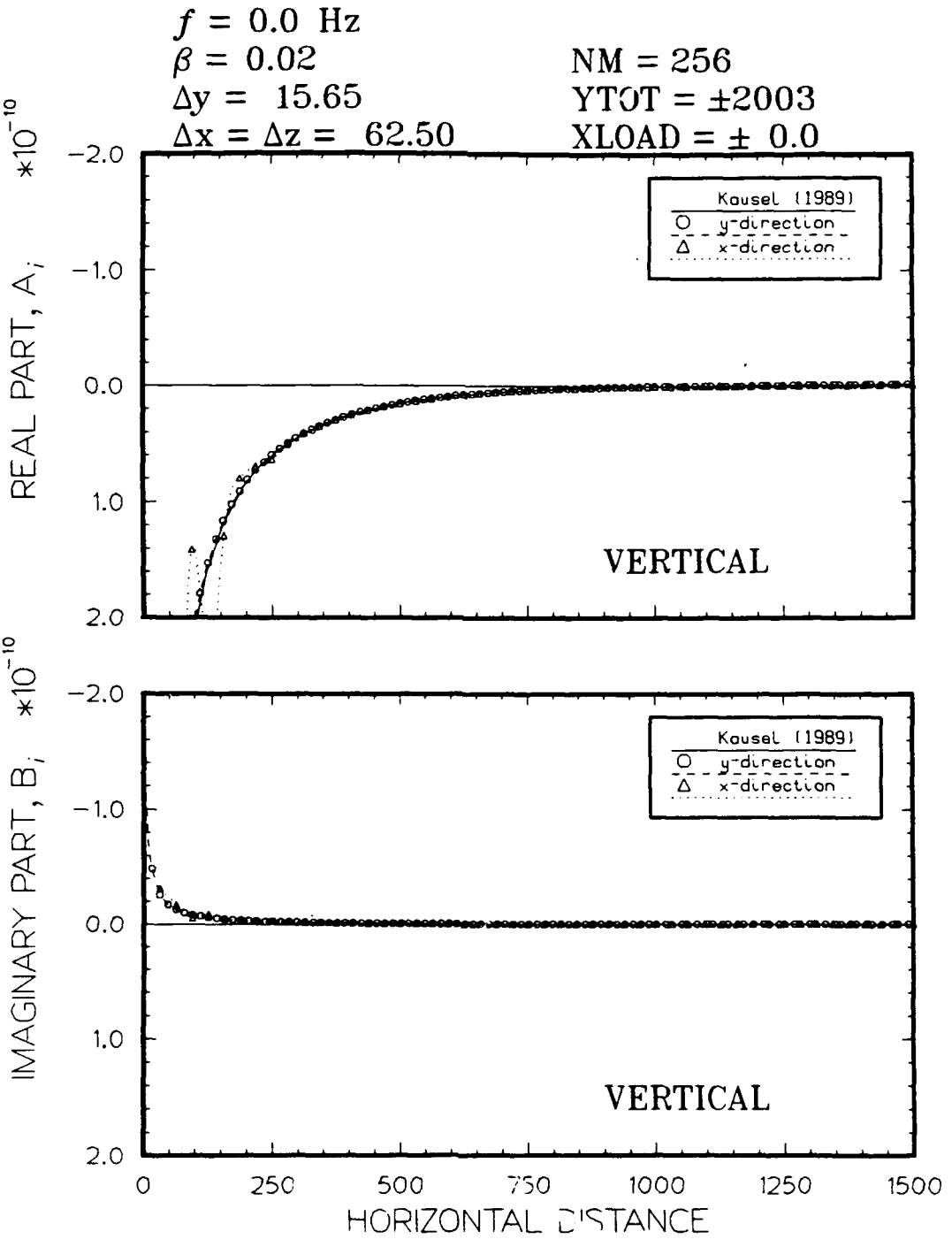


Figure 22. Vertical displacements at z = 0 for static point load and comparison with Green's function solution

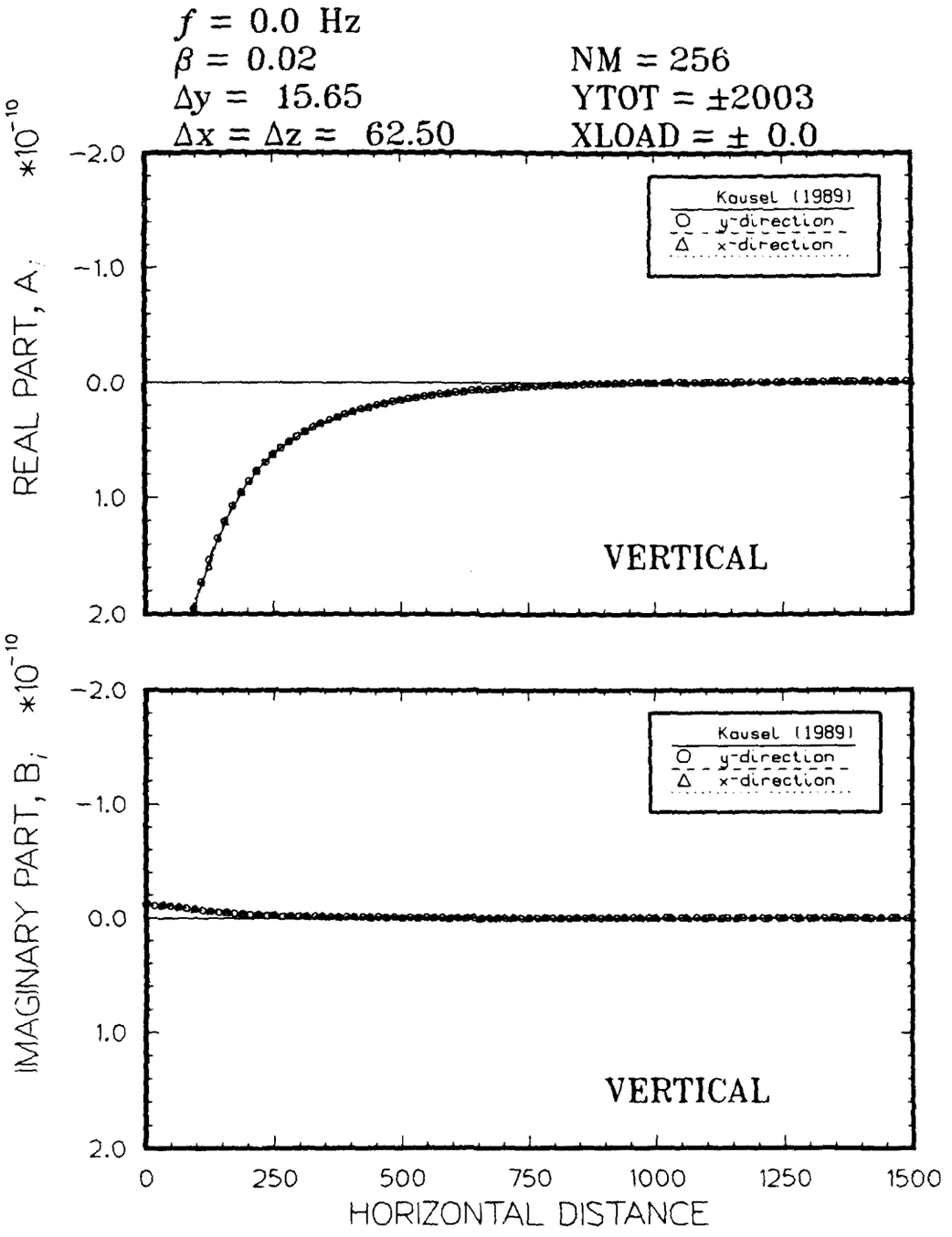


Figure 23. Vertical displacements at z = 125 for static point load and comparison with Green's function solution

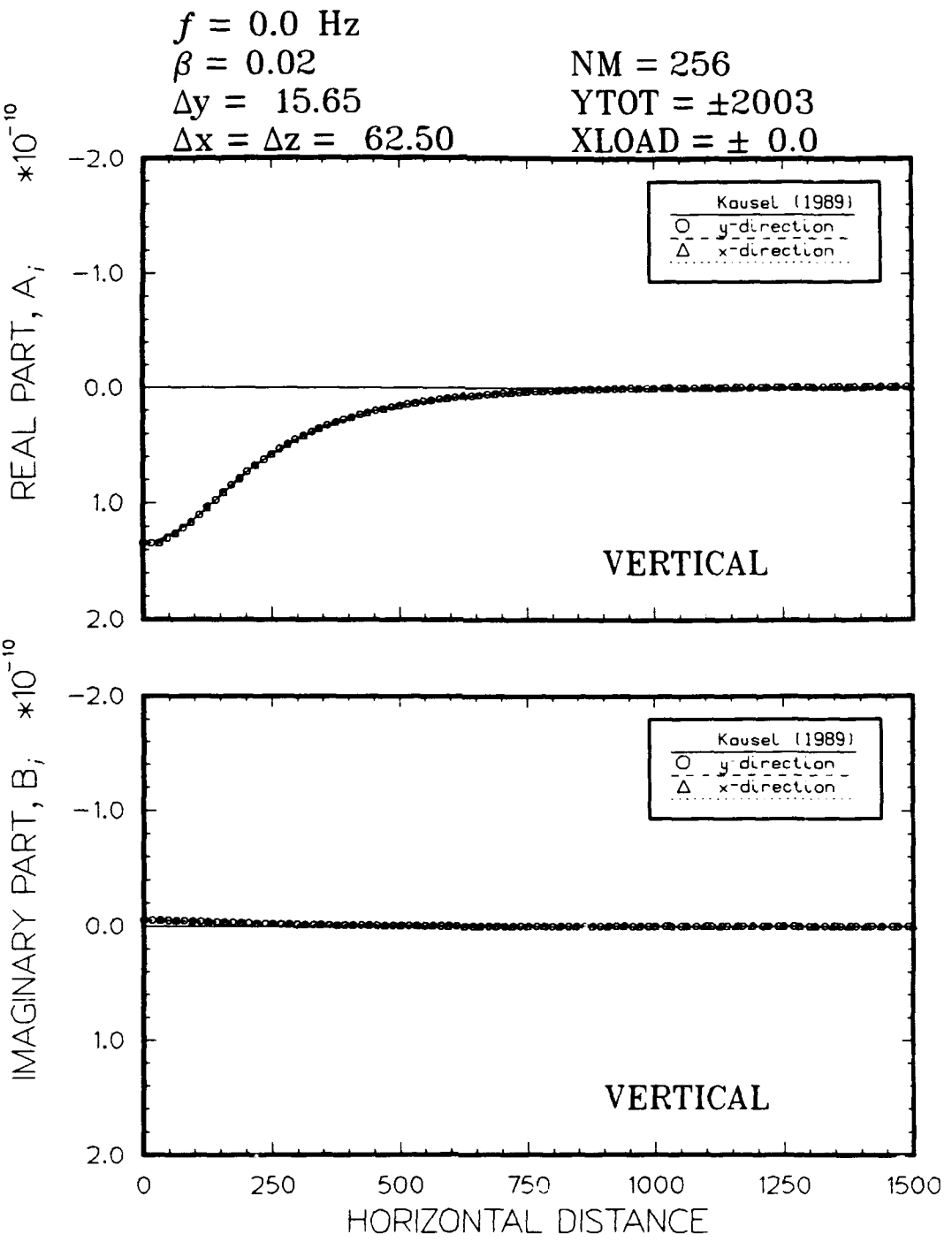


Figure 24. Vertical displacements at z = 250 for static point load and comparison with Green's function solution

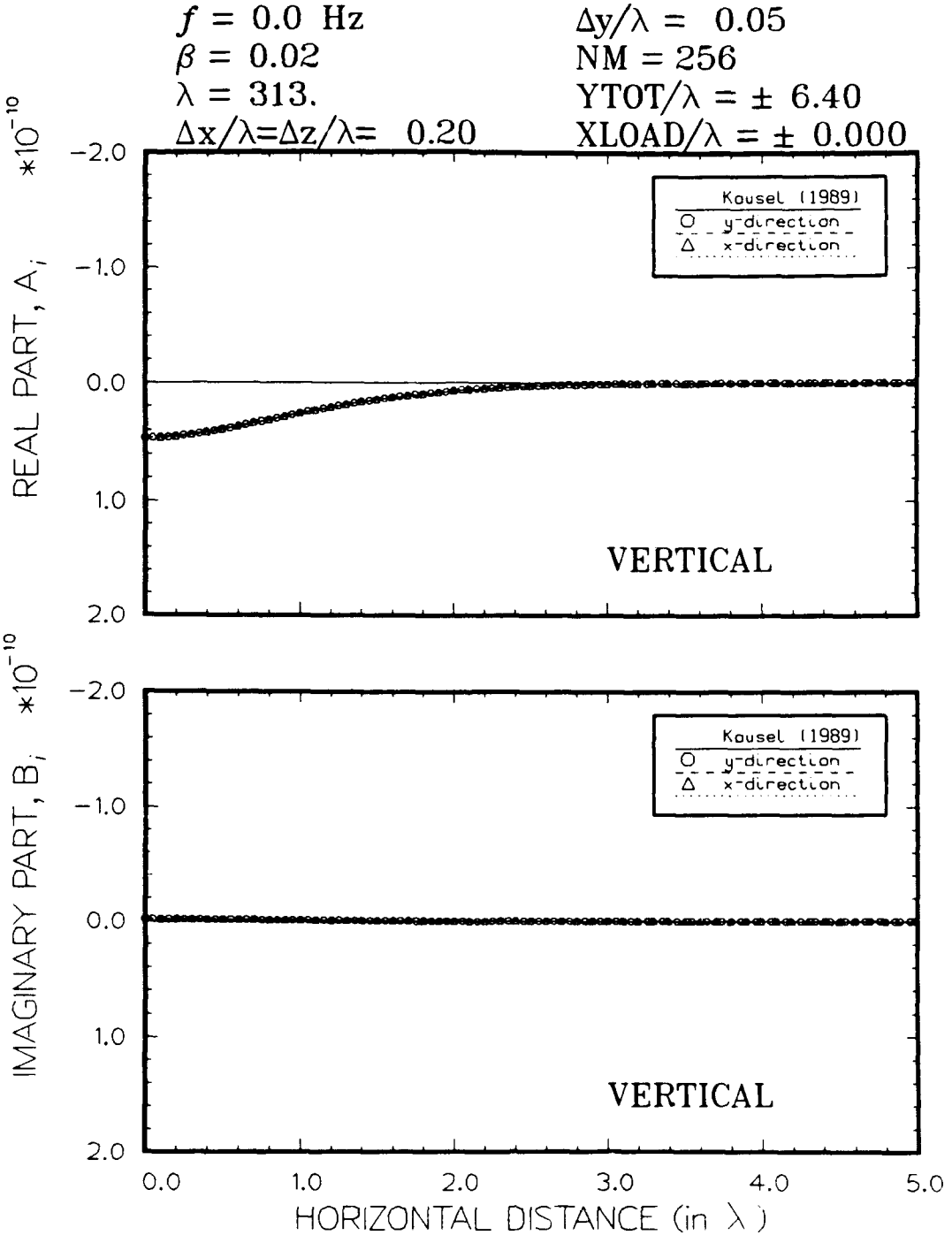


Figure 25. Vertical displacements at z = 500 for static point load and comparison with Green's function solution

were then compared with the Green's function solutions presented in the previous section. All four models were discretized using the finest mesh. A square load with plan dimensions of 5 by 5 centered about the origin with a total load of 1 was applied at a frequency of 3 Hz. The wavelength for Rayleigh waves is then about 313 and the dimension of the square elements are 62.5 units or about 0.2λ. The material properties are listed in Figure 16.

87. The parameters defining the condensation and Fourier expansion for the validation were selected based on the findings of Kang (1990). Values of $\Delta y = 0.05\lambda$ and the number of Fourier discretization points, NM, equal to 256 were fixed for the comparisons and the finest finite element mesh was used unless otherwise specified. This provided for a discretized extent (in the y-direction) of -13λ ($\pm 6.4\lambda$), slightly less than the total extent discretized in the x-direction ($\pm 8\lambda$). Displacements at distances up to 5λ, or about 1500 units, are used for comparison because the amplitudes are rather small beyond this distance.

Model 1: Homogeneous system

88. The results for Model 1 at $z = 0$ and $z = 0.40\lambda$ (125 units) are shown in Figures 26 and 27, respectively, and compared with the Green's function solutions. The variation of the real part of the complex displacements in the y-direction compares favorably to the Green's function solution. The variation of the imaginary part in the y-direction closely follows the Green's function solutions. Both parts of the calculated solutions compare more favorably at distances less than 3λ, about half the distance expanded in the y-direction. The comparisons are also slightly better at $z = 0.4\lambda$ as compared to $z = 0$. The variation of vertical displacements in the x-direction at $z = 0$ and $z = 0.40\lambda$ generally compare favorably. These displacements differ somewhat from the variation in the y-direction with the imaginary part deviating more.

89. The effects of varying Poisson's ratio and damping ratio on the displacements for Model 1 were also examined and the results are shown in Figures 28 and 29, respectively. A comparison between the calculated and the Green's function solutions in Figure 20 for variations in ν indicates that the 2-D approximation provides a reasonable means of representing different ν. The best comparison is for $\nu = 0.40$ and the poorest comparison is for $\nu = 0.49$. Generally, the imaginary part compares well and the real part has amplitudes that are consistently too small. The 2-D approximation provides an

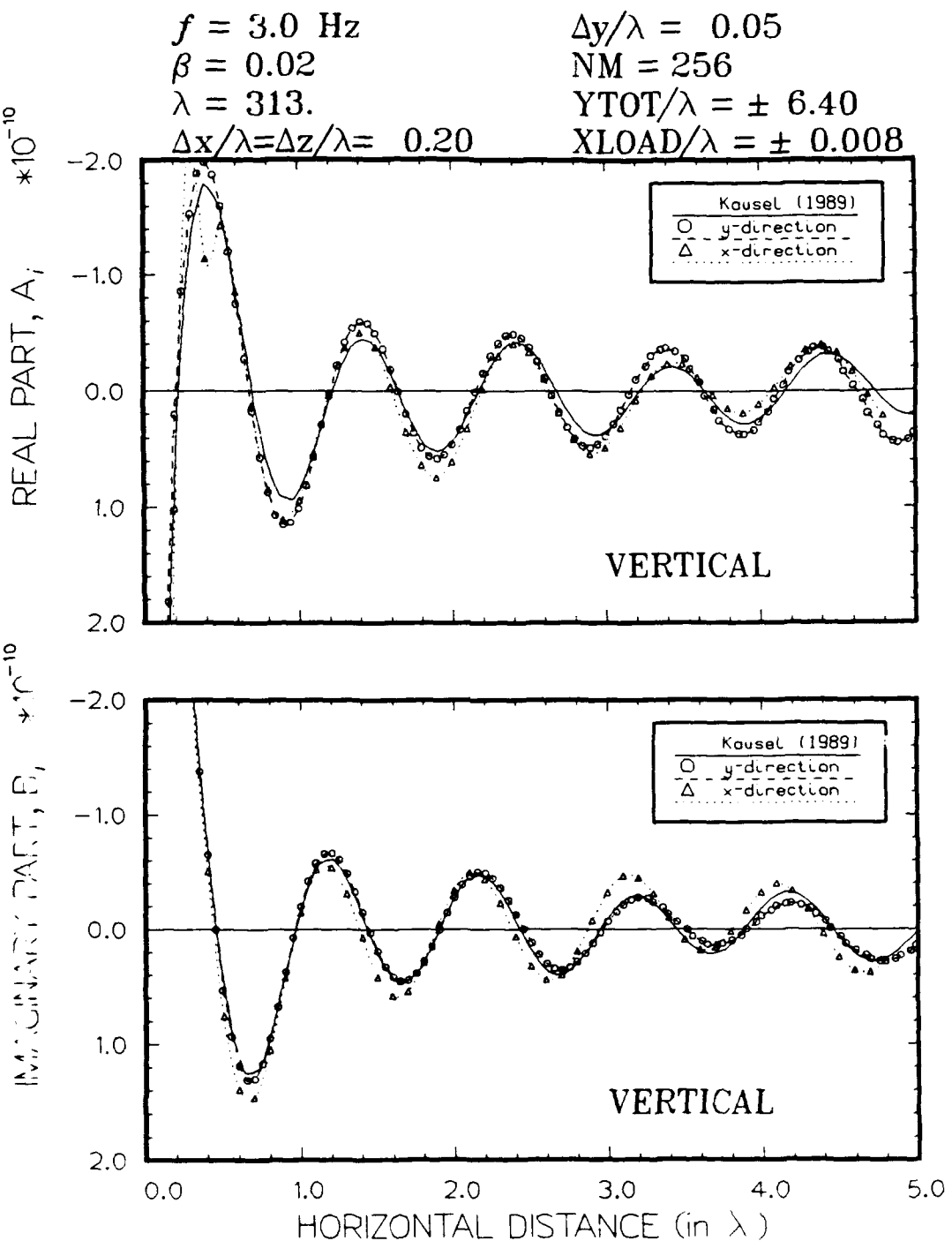


Figure 26. Comparison of dynamic displacements at z = 0.0 for Model 1 with ν = 0.40

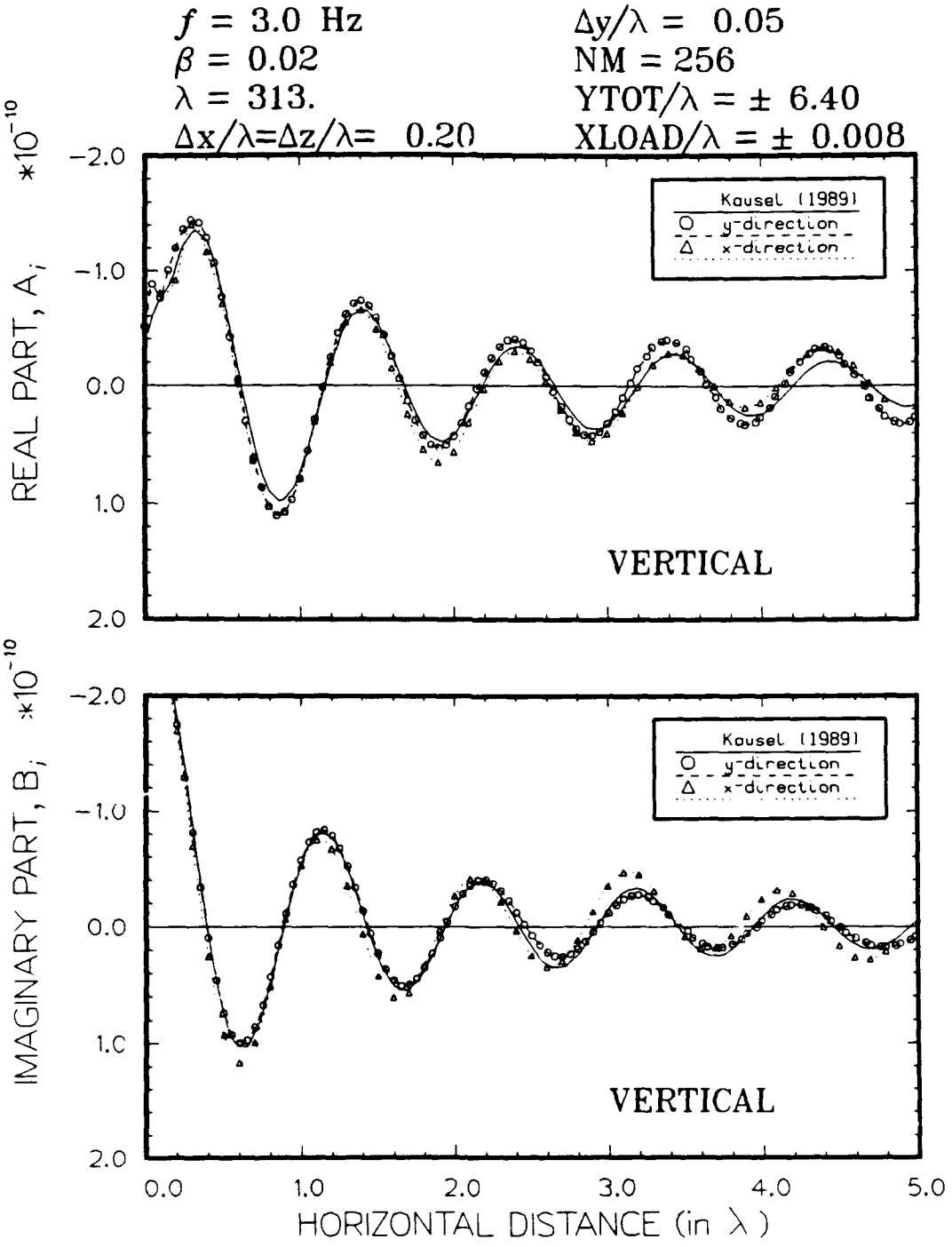


Figure 27. Comparison of dynamic displacements at z = 0.40λ for Model 1 with ν = 0.40

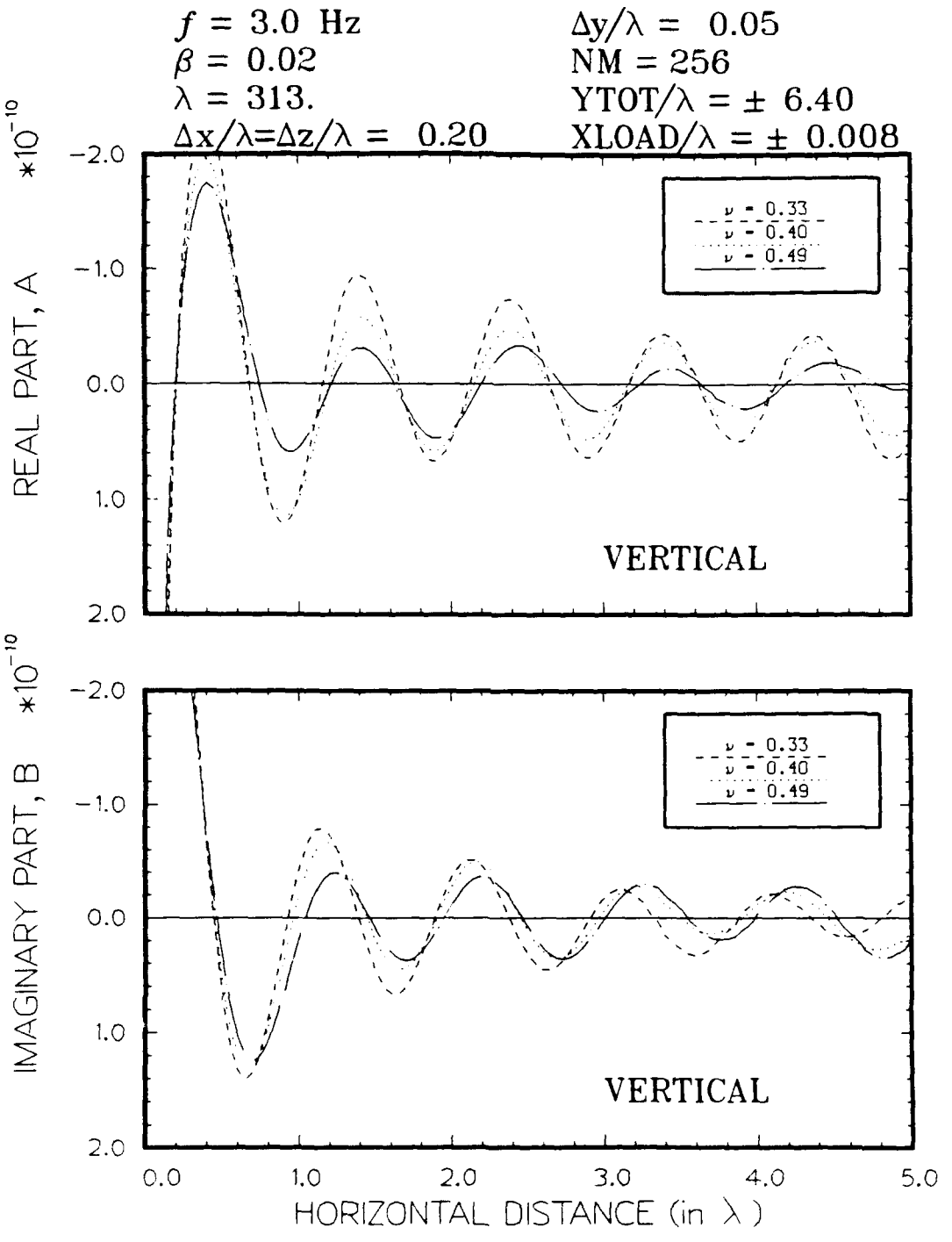


Figure 28. Comparison of dynamic displacements for Model 1 using various Poisson's ratios

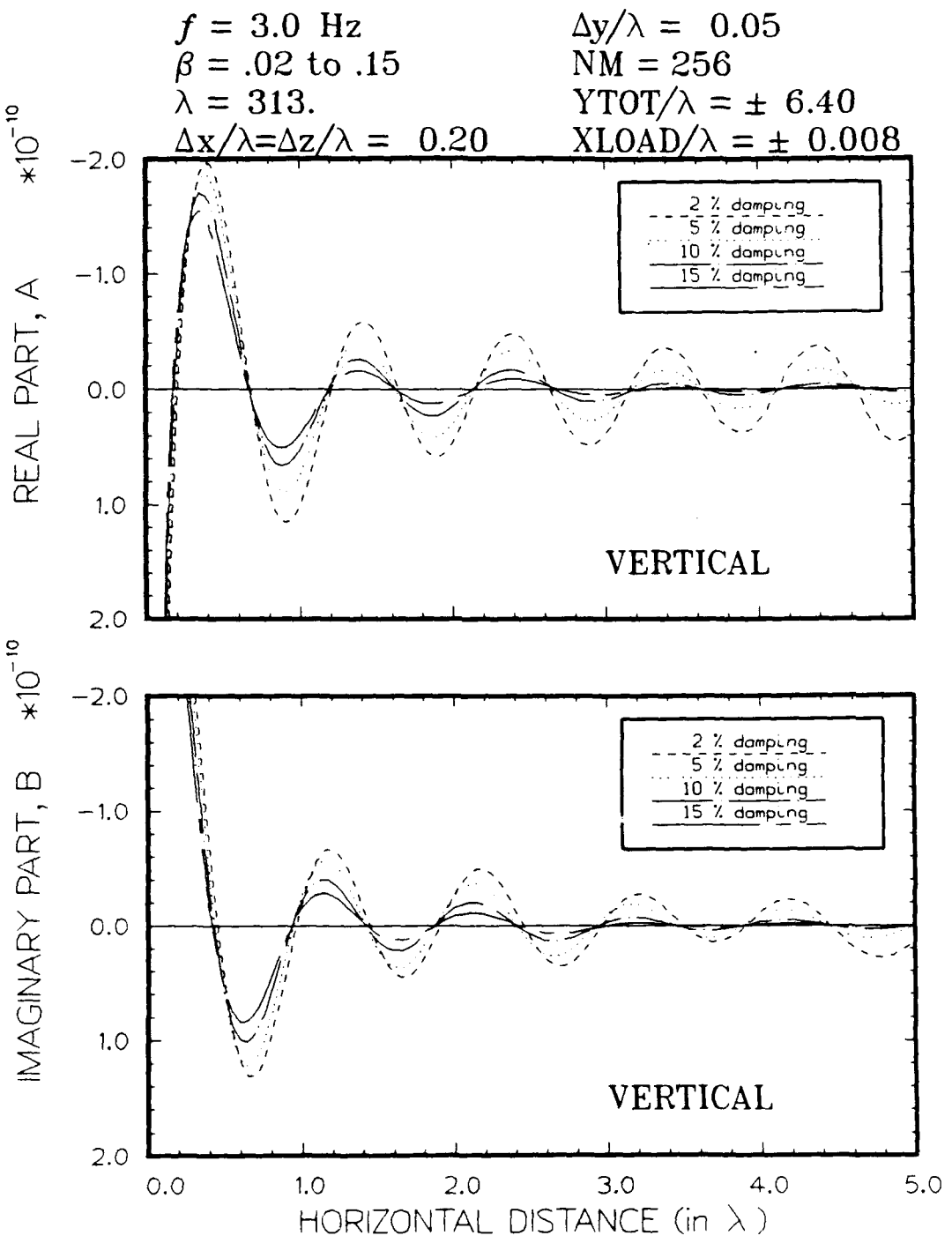


Figure 29. Comparison of dynamic displacements for Model 1 using various damping ratios

accurate means of representing damping as seen by comparing relations in Figures 21 and 29. The accuracy of calculated displacements improves somewhat as the damping ratio increases. The results for 5 percent damping compare much better with the Green's function solution than the results for 2 percent. Both 2 and 5 percent damping levels are used for comparisons hereafter.

Other models: Stiffness varying with depth

90. The results for Models 2, 3 and 4 are shown in Figures 30 through 32 and compared with the Green's function solutions. Total distances are used rather than normalized distances since considerable dispersion is expected. The results for vertical displacements in the x- and y-direction are nearly equivalent to the Green's function solutions except at the first peak in the real part for all three cases.

Conclusions

91. Comparisons made between vertical displacements calculated using *vib3* and Green's function solutions (Kausel 1981; Kausel 1989) for the simple case of layered axi-symmetric geosystems suggest that the formulation for the specialized element and the implementation in *vib3* are sound, accurate, and stable. Comparisons were made for static and dynamic loads and for reasonable ranges of Poisson's ratio and damping ratio. Both of these factors were found to have significant effect on dynamic displacements which emphasizes the importance of obtaining adequate values. The calculated displacements were shown to be more accurate as damping ratio was increased from 2 to 5 percent. The variations of dynamic vertical displacements in the x- and y-directions generally differ reflecting differences in interpolation orders, spatial discretization, and possibly other effects. More specific examinations of these differences are contained in Part V.

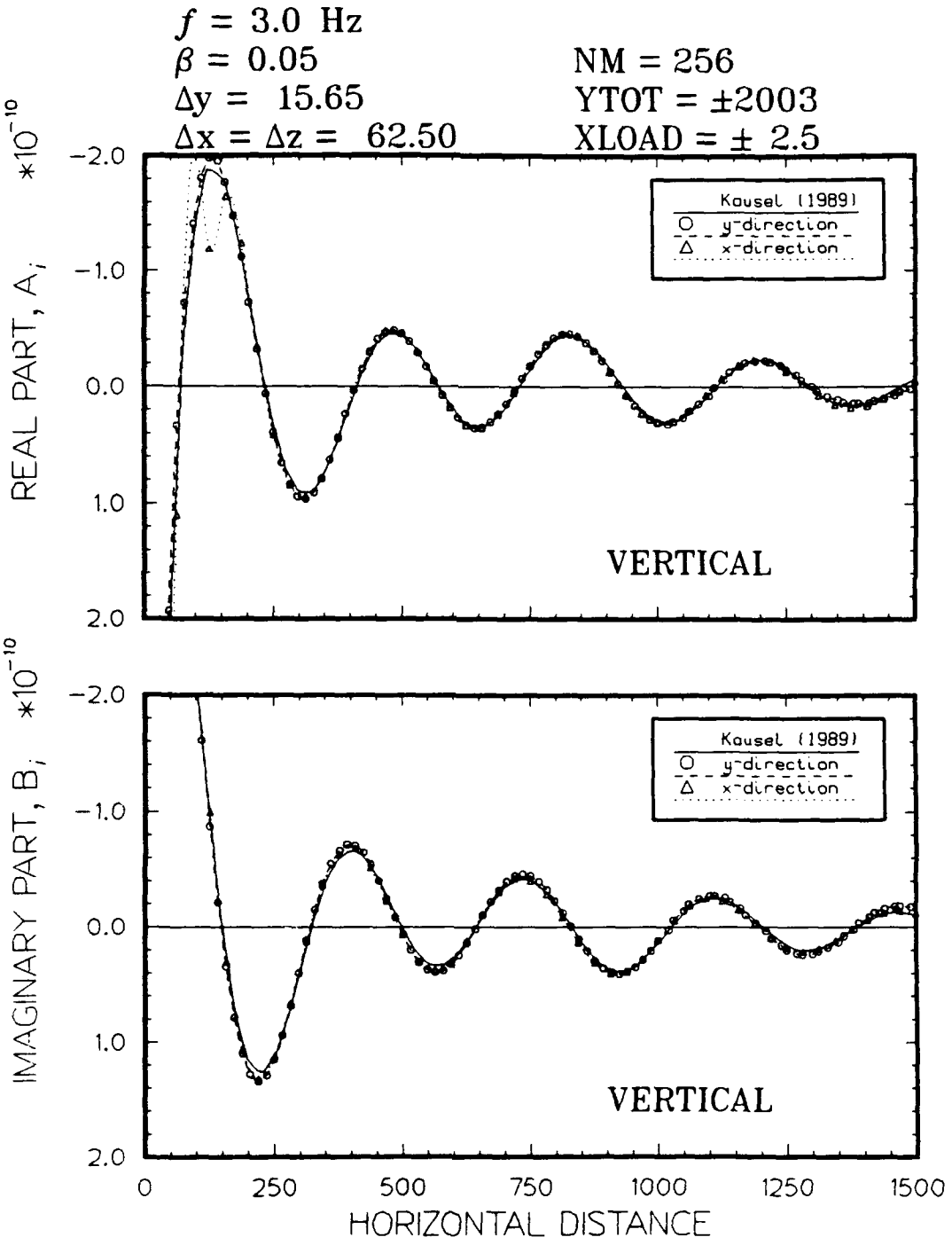


Figure 30. Vertical displacements for Model 2

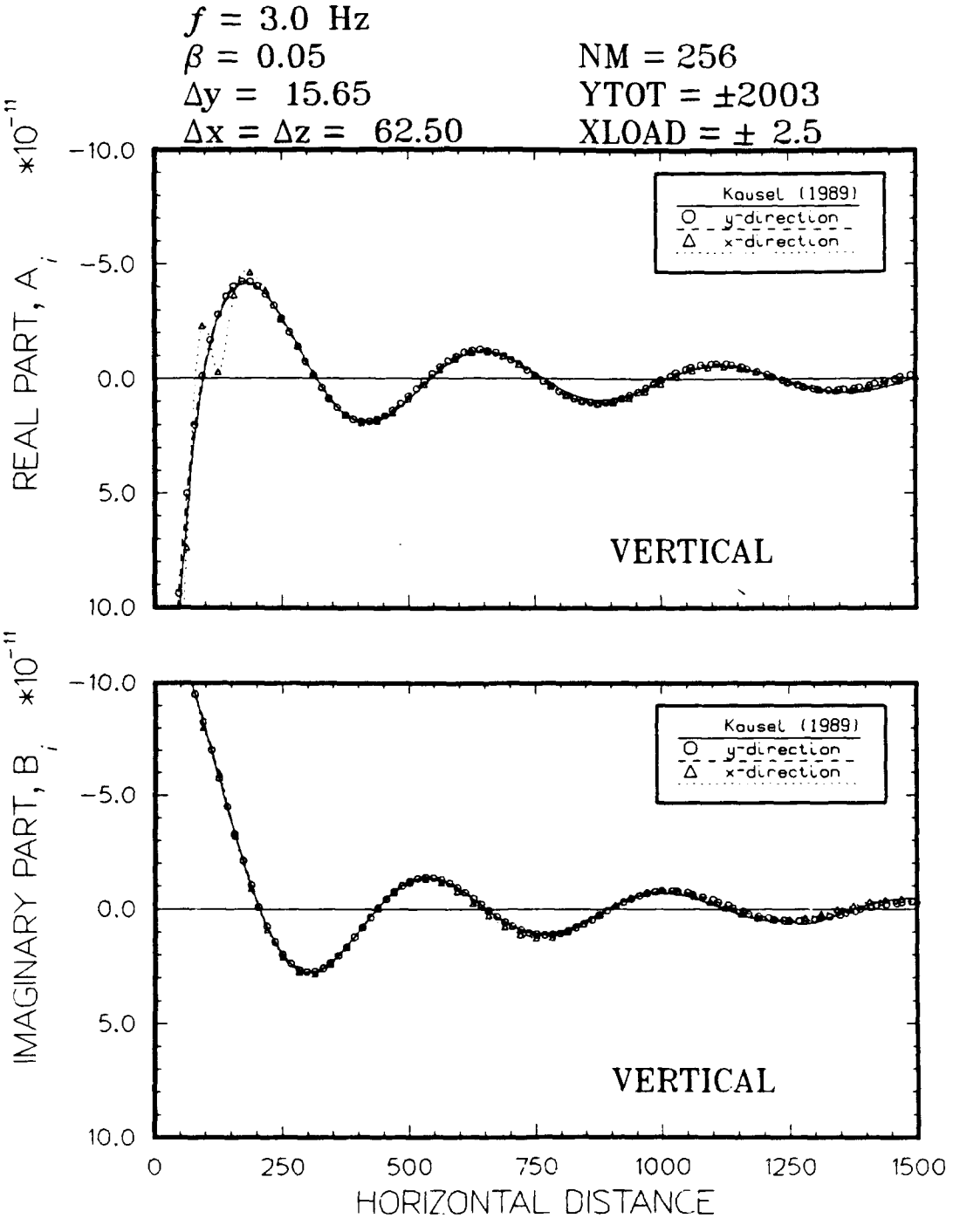


Figure 31. Vertical displacements for Model 3

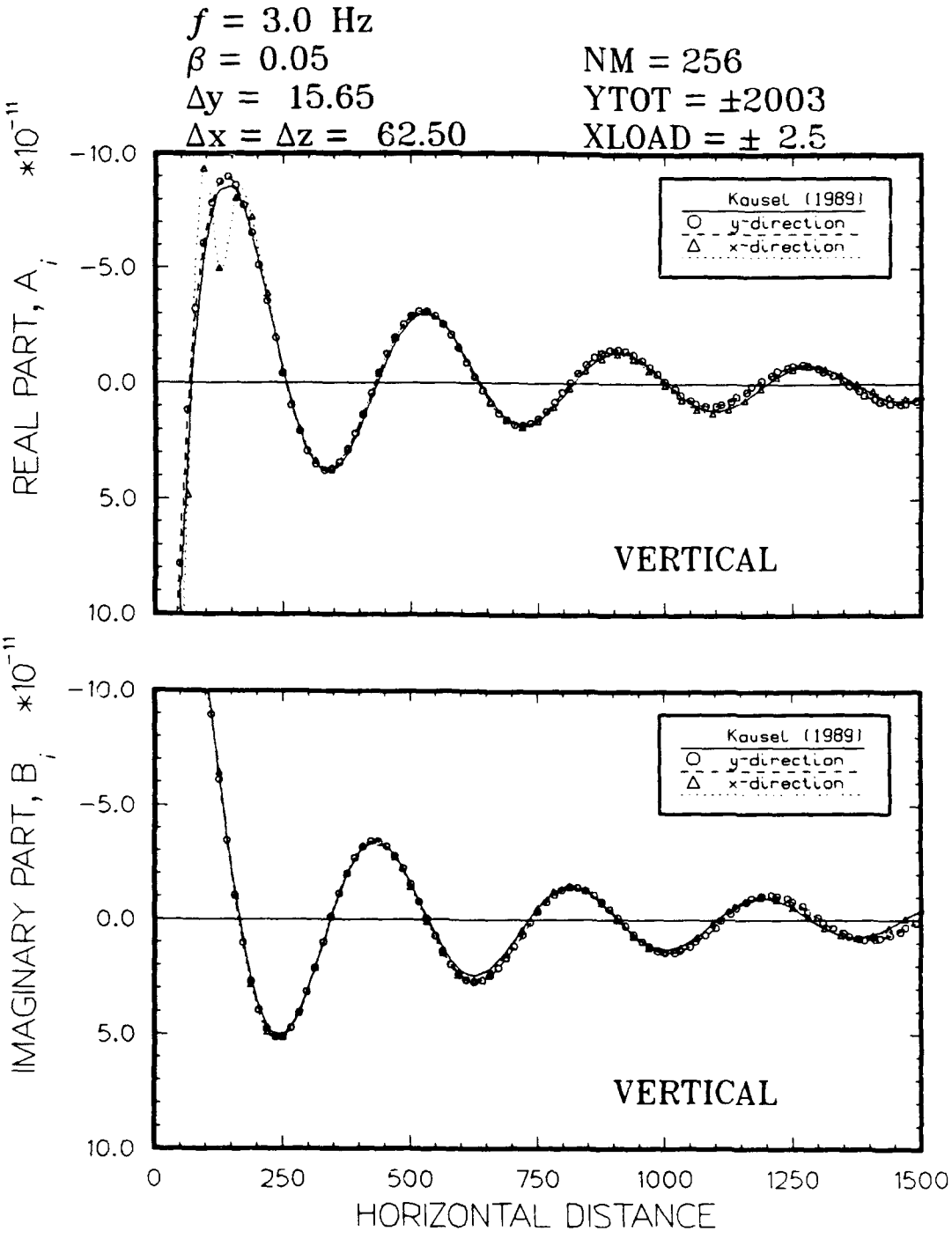


Figure 32. Vertical displacements for Model 4

PART V: PARAMETRIC ANALYSES

92. Parametric analyses were conducted to assess the sensitivity of the formulation and computer code *vib3* to anticipated ranges of system variables. Calculations were made using Model 1 and the finest mesh (refer to Figures 16 and 17) except in the case of examining sensitivity to mesh size. Green's function solutions calculated using *PUNCH* (Kausel 1989) and presented in Part IV are used for comparison. Some minor differences may be expected for comparisons between the Green's function solutions and the 2-D approximations since the shape of the load is different; the same total area (and load) were used to minimize these differences.

Effect of Δy

93. The effect of the spatial increment of discretization in the y-direction was evaluated by comparing the results using three values of Δy between 0.05λ and 0.20λ (0.05λ used for validation in Part IV). The number of FFT points, NM, was also varied to keep the total discretized distance in the y-direction, YTOT, constant. This distance is defined by:

$$YTOT = NM \cdot \Delta y \tag{78}$$

Keeping YTOT constant serves to isolate the effects of Δy. The dynamic displacements for each Δy are presented in Figures 33 through 35 along with the Green's function solution. The variation of vertical displacements in the y-direction among the values of Δy are compared in Figure 36.

94. The large difference among relationships presented in Figure 36 indicate that Δy has a significant effect on the ability of *vib3* to accurately calculate dynamic displacements. Comparisons between the calculated displacements and Green's function solutions are favorable when $\Delta y \leq 0.10\lambda$ although some improvement is noticeable by decreasing Δy to 0.05λ. These results are consistent with Kang's (1990) who recommended that $\Delta y \leq 0.10\lambda$. The results for $\Delta y \geq 0.20\lambda$ are considered to be too inaccurate. The calculated variations of vertical displacements in the x-direction are very similar for all Δy which suggests that the solution in the x-direction is insensitive to Δy within the range of values considered.

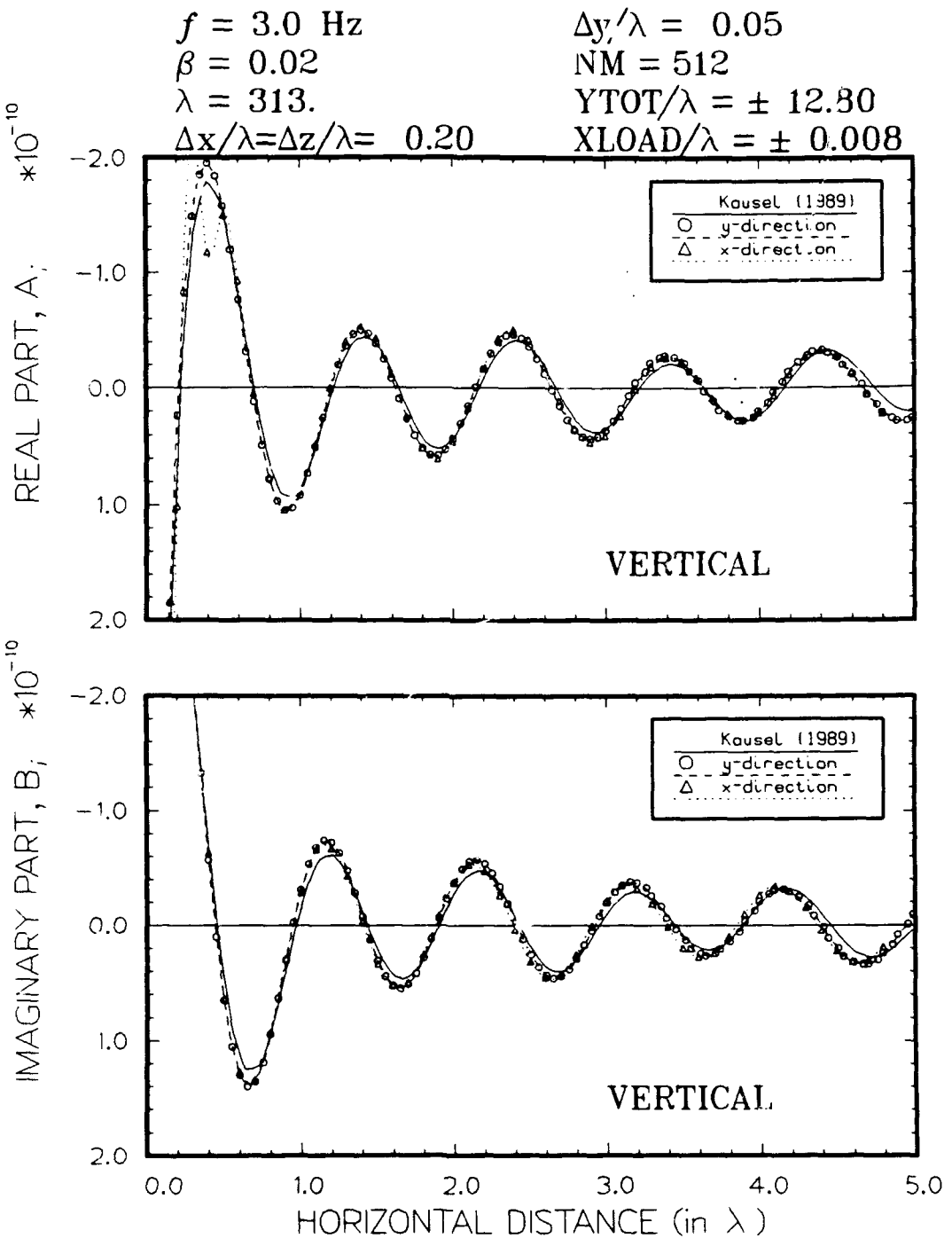


Figure 33. Vertical displacements at Δy = 0.05λ and NM = 512 and Green's function solution

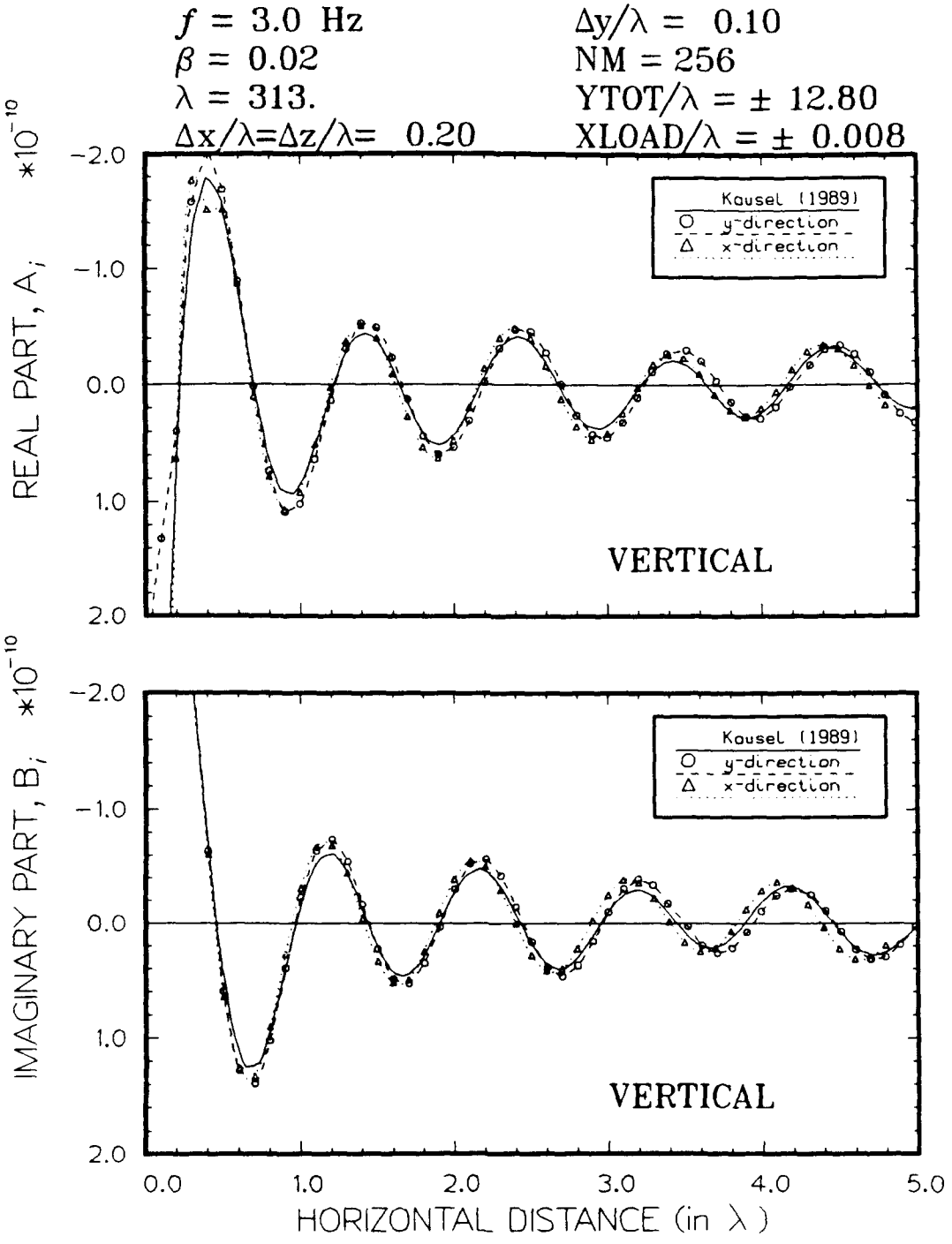


Figure 34. Vertical displacements at Δy = 0.10λ and NM = 256 and Green's function solution

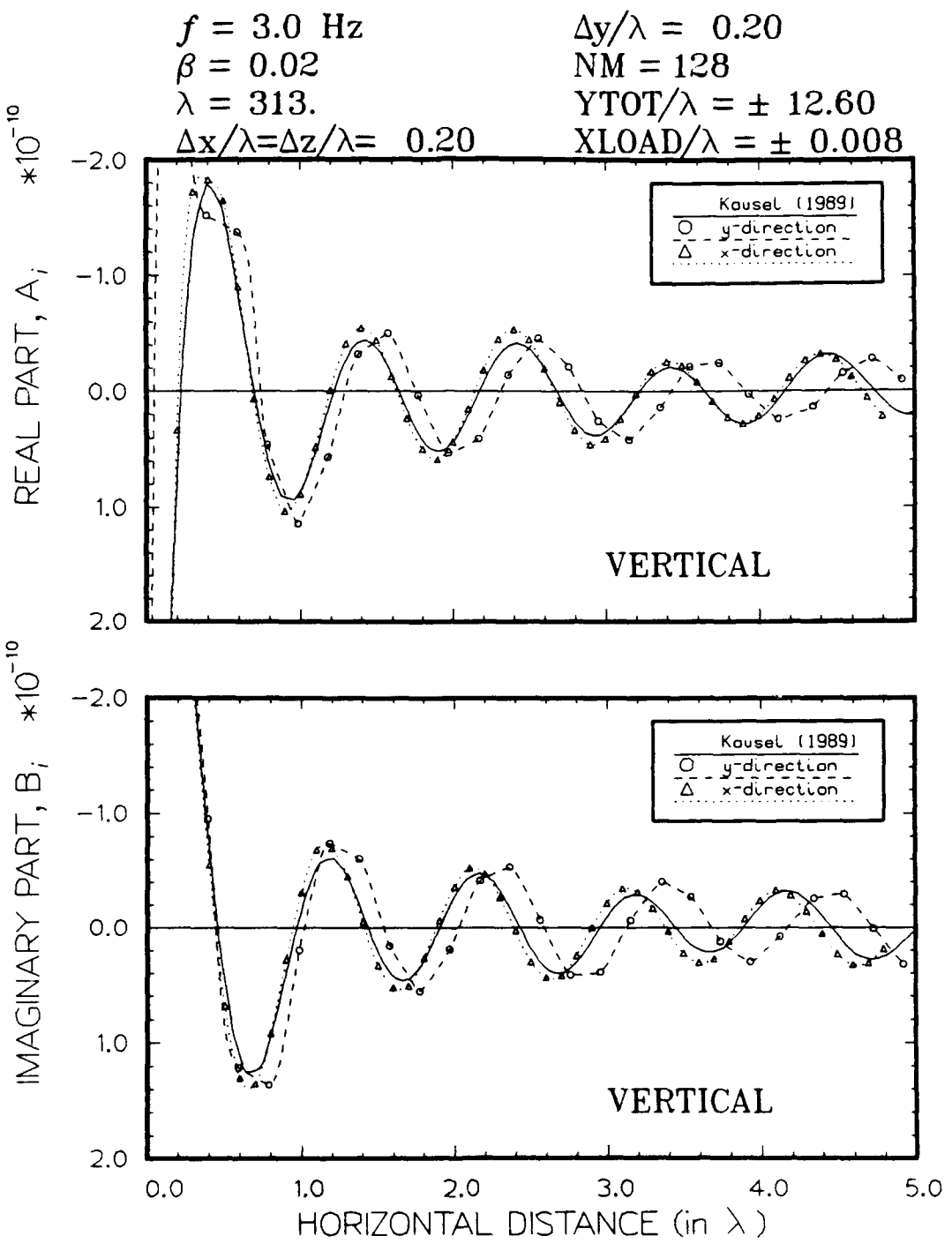


Figure 35. Vertical displacements at Δy = 0.20λ and NM = 128 and Green's function solution

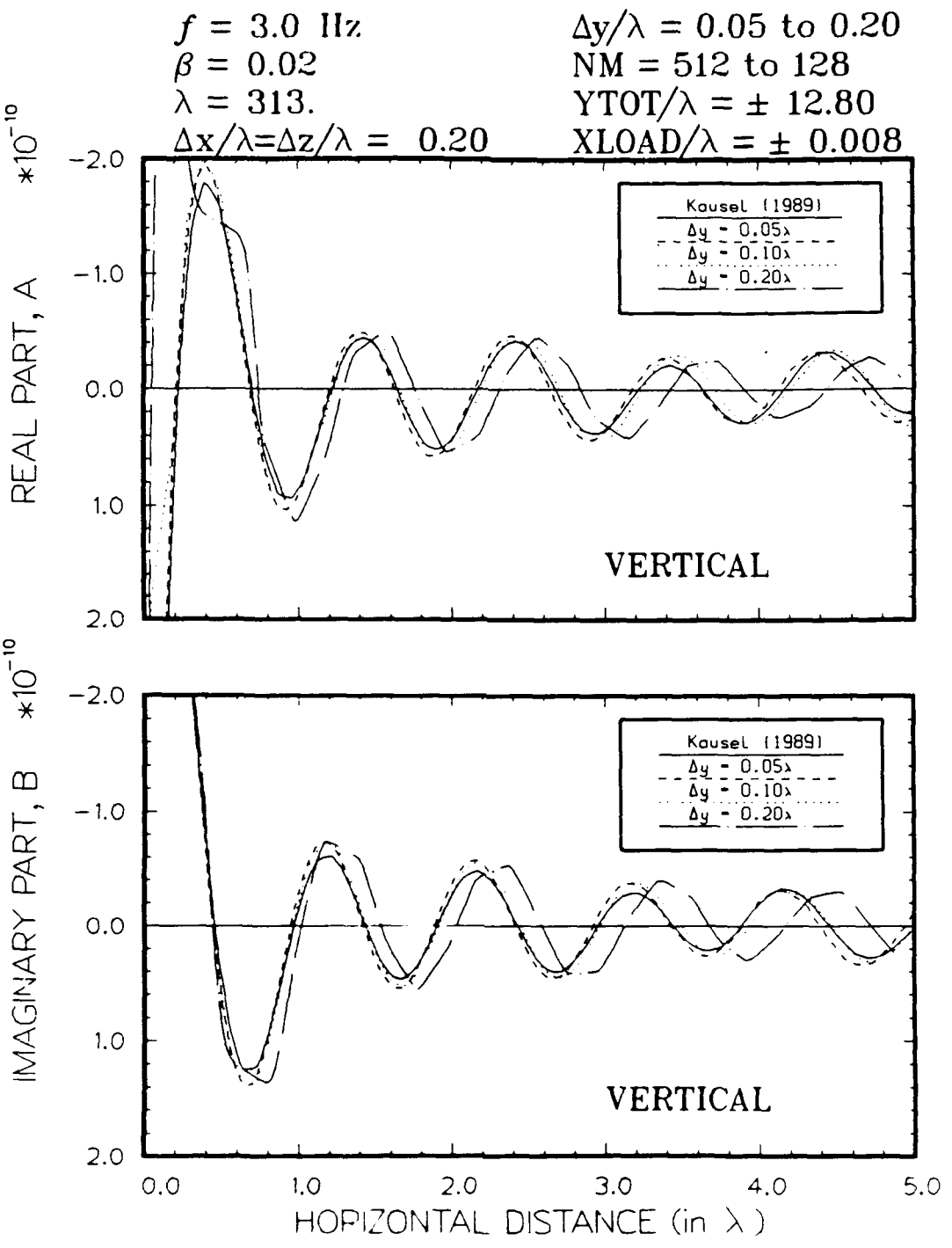


Figure 36. Comparison of vertical displacements showing effect of Δy with Green's function solution

95. The parametric analysis of Δy with respect to λ indirectly addresses the effect of frequency of excitation on the results. For a homogeneous system (with constant stiffness), λ is inversely proportional to frequency. So, the spatial increment Δy can also be put in terms of frequency:

$$\Delta y \leq \frac{V}{10\, f} \tag{79}$$

where

V = phase velocity
f = frequency (Hz)

The phase velocity can be taken equal to the Rayleigh wave velocity as a first approximation. Similar relationships to Equation 79 have been observed in other types of discretized solutions for dynamic loading.

Effect of Extent of Fourier Expansion

96. The comparisons for the effect of Δy were made using a constant value of YTOT. The effect of varying YTOT was examined next. The total distance was varied at three values between 3.2λ and 12.8λ (corresponding to $\pm\ 1.6\lambda$ and $\pm\ 6.4\lambda$, respectively) by keeping Δy constant at 0.05λ and varying NM between 128 and 512. The dynamic displacements for each YTOT are presented in Figures 37 through 39 along with the Green's function solution. The variation of vertical displacements in the y-direction for the three values of YTOT are compared in Figure 40. The results for the vertical displacements are very good for the case of YTOT = $\pm\ 12.8\lambda$. The results for YTOT = $\pm\ 6.4\lambda$ are also good, especially for distances less than 3λ, and the results for YTOT = $\pm\ 3.2\lambda$ are considered to be too inaccurate. A threshold of 10λ is likely to be appropriate.

97. The variation of vertical displacements in the x-direction improves as YTOT increases. The most significant improvement occurred as YTOT increased from $\pm\ 3.2\lambda$ to $\pm\ 6.4\lambda$. The solution in the x-direction was shown to be independent of Δy so the dependence can be attributed to either NM or YTOT. The dependency is most likely caused by YTOT resulting from reflections off the free end(s) in the y-direction. Less reflected energy returns to the $y = 0$ plane as the total discretized length in the y-direction increases. The threshold for YTOT defined for the solution in the y-direction appears to be

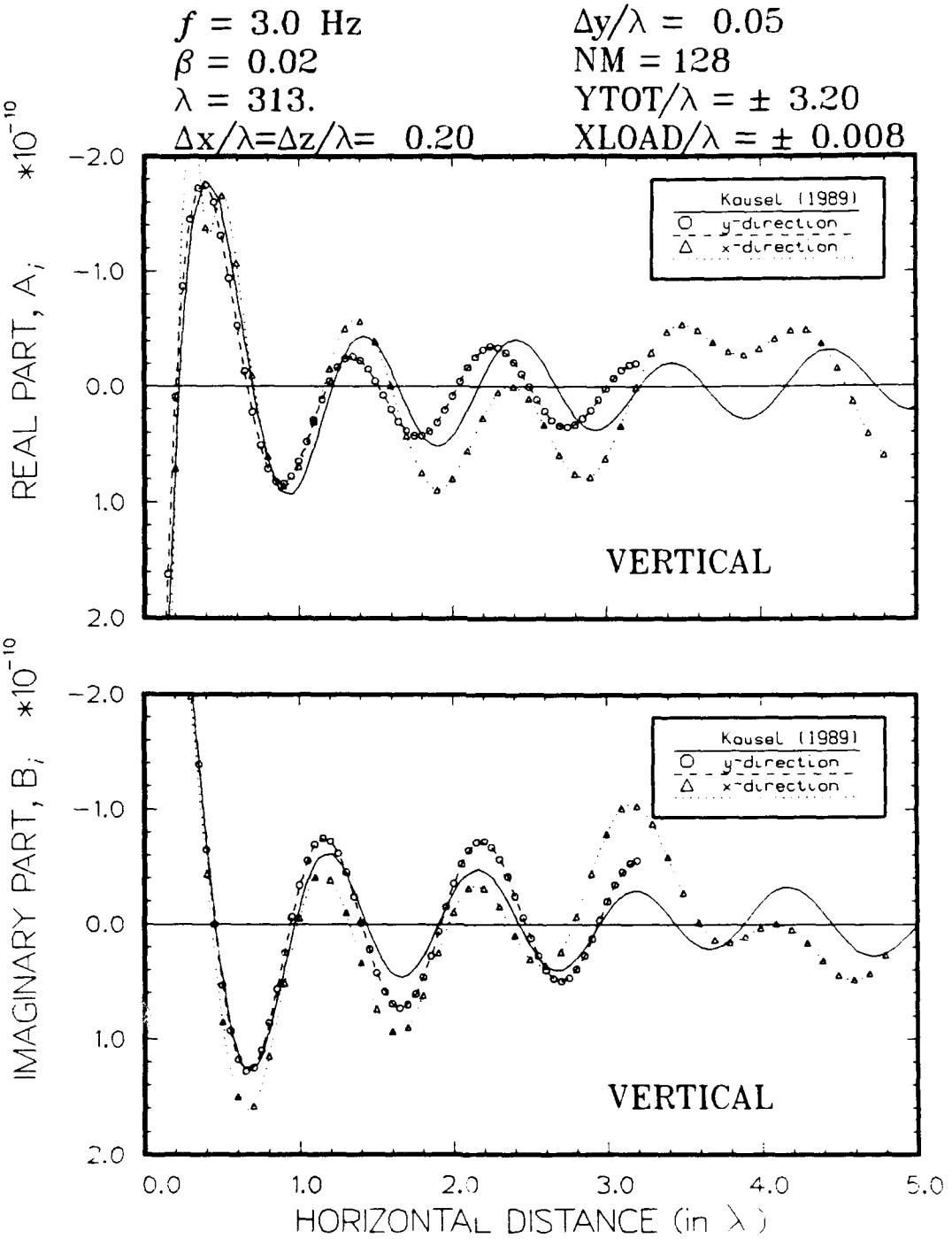


Figure 37. Vertical displacements at YTOT = ± 3.2λ and Green's function solution

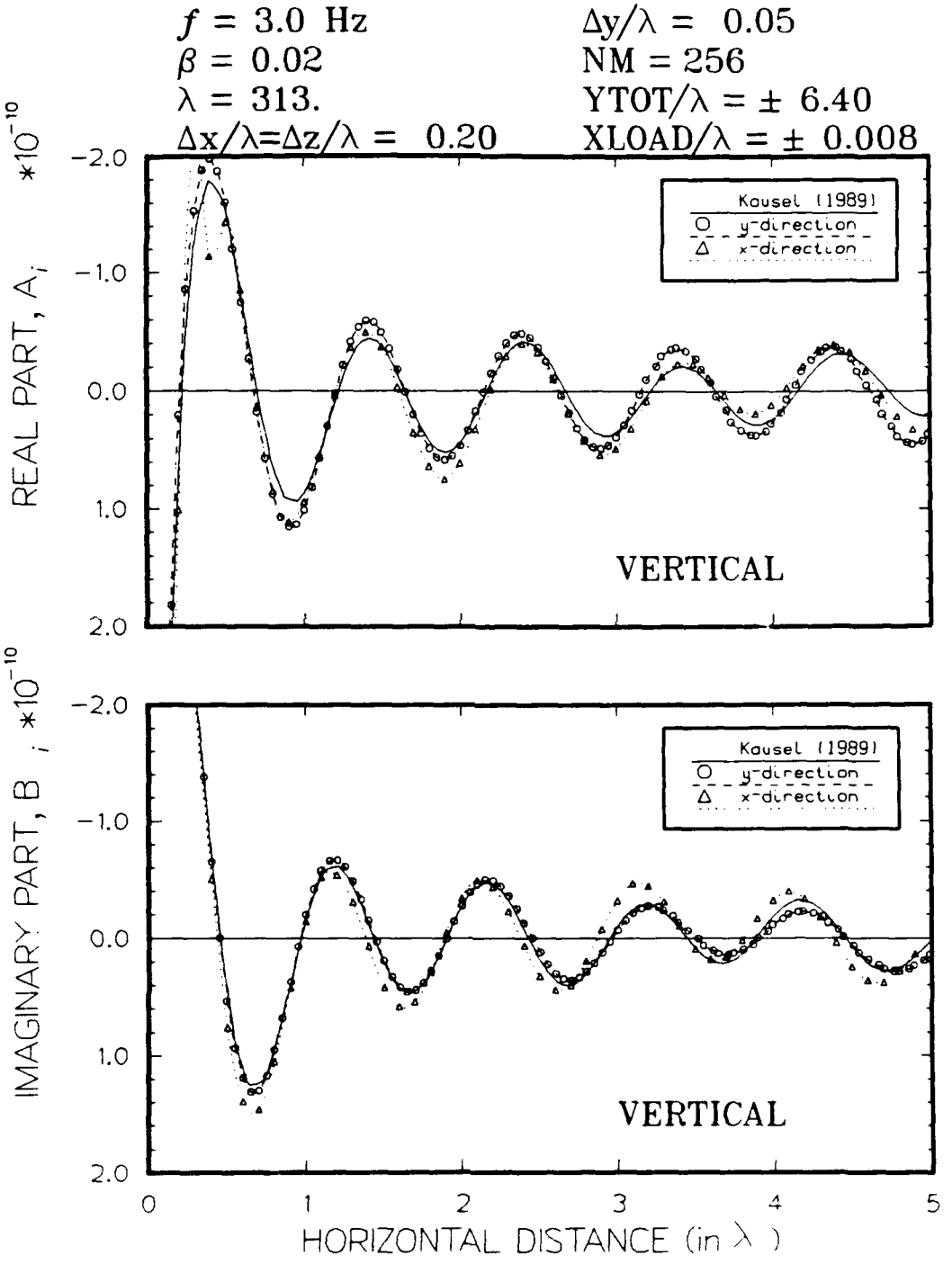


Figure 38. Vertical displacements at YTOT = ± 6.4λ and Green's function solution

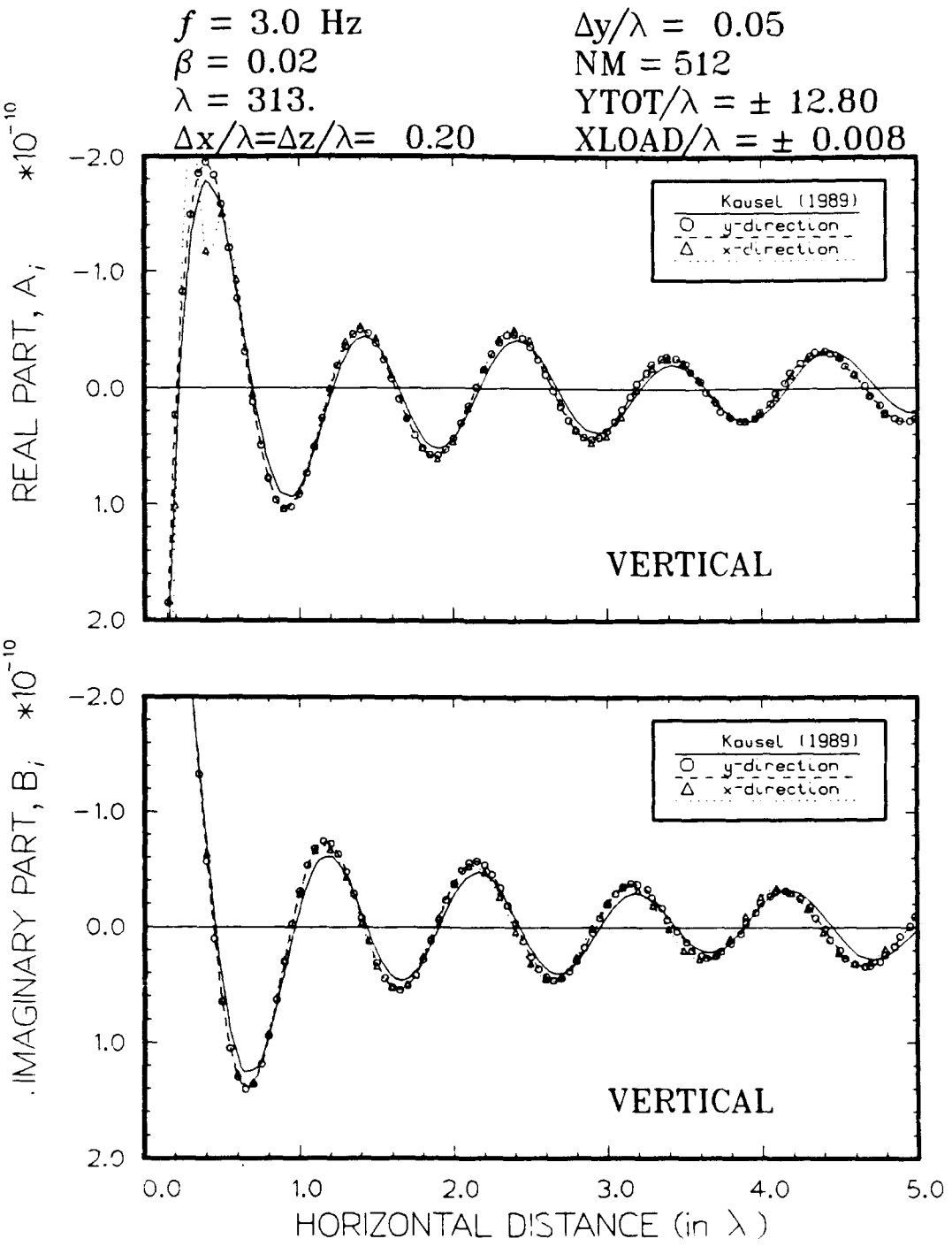


Figure 39. Vertical displacements at YTOT = ± 12.8λ and Green's function solution

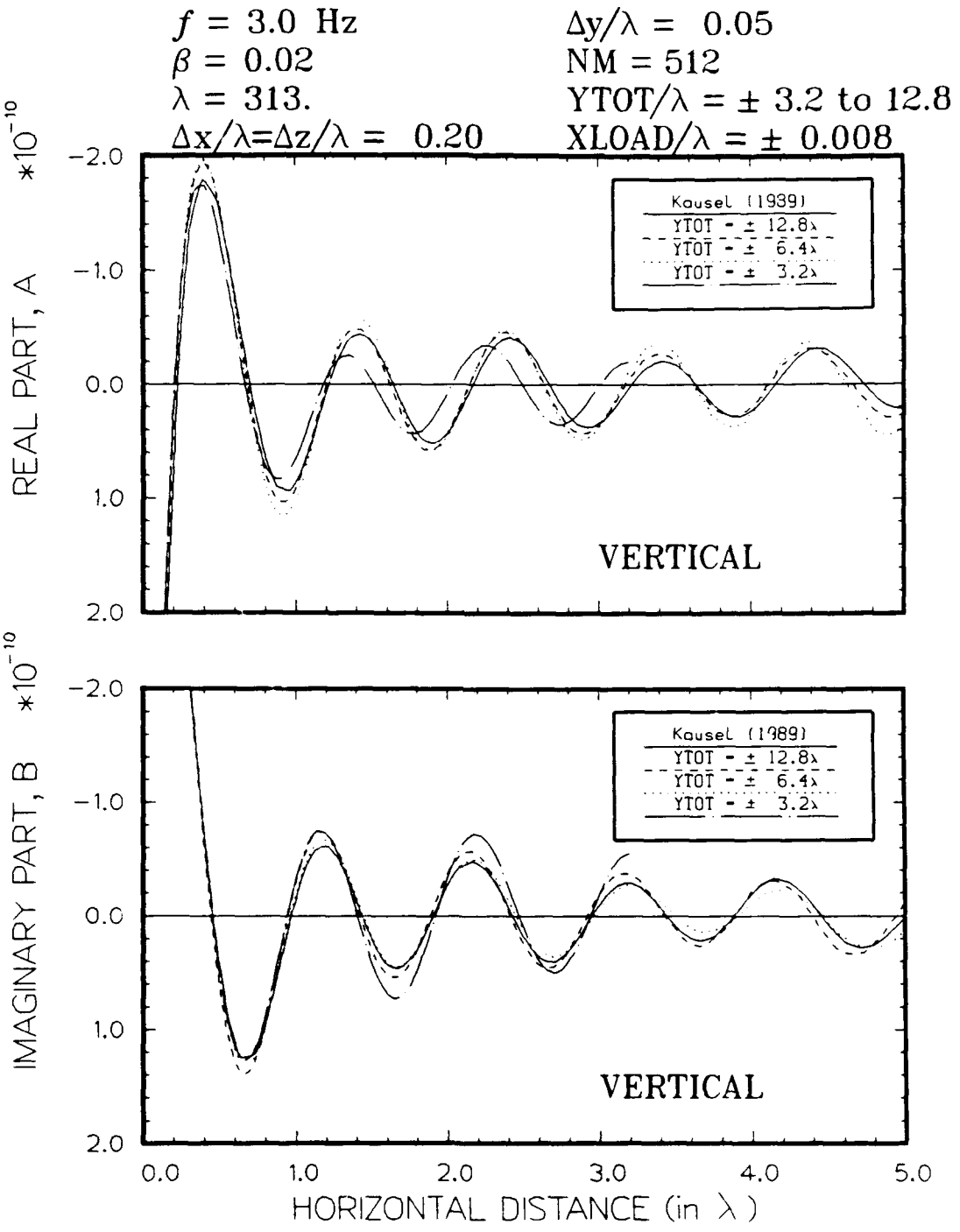


Figure 40. Comparison of vertical displacements showing effect of YTOT with Green's function solution

suitable for the solutions in the x-direction based upon the results shown in Figures 37 through 39.

Effect of Element Size

98. The effect of varying the size of the finite elements on the dynamic displacements was determined by using the three different meshes shown in Figure 17. The values of Δx (= Δz) corresponding to these three meshes are 0.20λ, 0.40λ, and 0.80λ. The vertical displacements for the finest mesh was presented previously in Figure 26 and the displacements for the other two meshes are shown in Figures 41 and 42 along with the Green's function solution. The results for the variation in the y-direction for the three meshes are compared in Figure 43.

99. The variation of vertical displacements in the y-direction compare well with the Green's function solutions except for the coarsest mesh (4 by 10 elements). The results for the coarsest mesh are unacceptable. The finest mesh produces peak values of displacement slightly greater than the Green's function solution and the original mesh. (This solution may be more accurate than the Green's function solution which had not entirely converged.)

100. The variation of vertical displacements in the x-direction also compare well with the Green's function solutions except for the displacements corresponding to the coarsest mesh which are unacceptable. The solutions in the x-direction are generally different from the solutions in the y-direction and deviate slightly more from the Green's function solution. The real part of dynamic displacements for the finest mesh shown in Figure 26 has an anomalous inversion at the first peak which tends to occur only for displacements in the x-direction calculated using the finest mesh.

101. The variations of displacements in both the x- and y-direction are dependent on the discretization in the x-direction. The dimensions of finite elements with quadratic interpolation functions can be 0.40λ although some improvement is expected as Δx is decreased further. This value is consistent with the conclusion of Kang (1990). A better threshold is probably equal to 0.30λ.

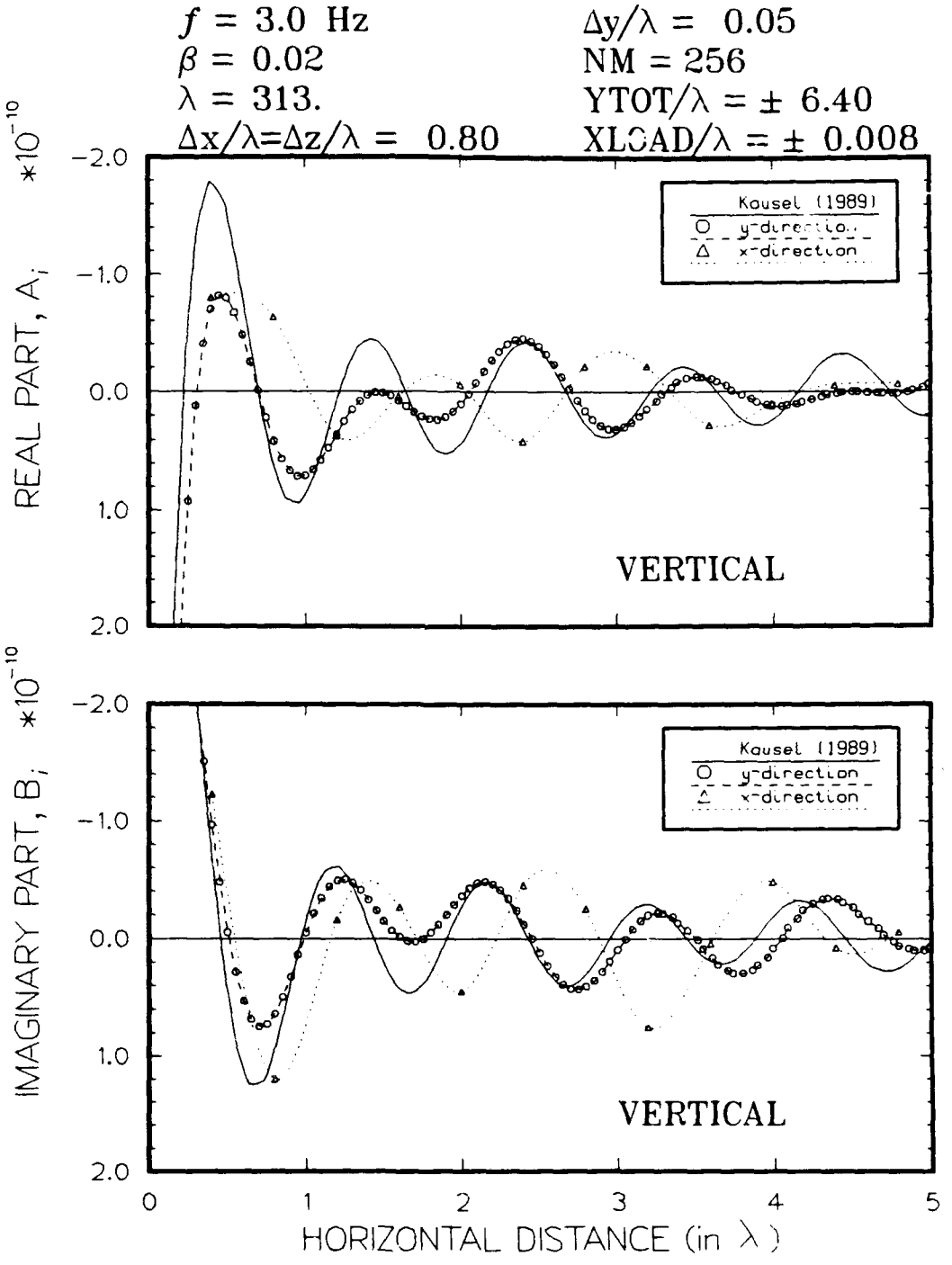


Figure 41. Vertical displacements discretized with 40 elements and Green's function solution

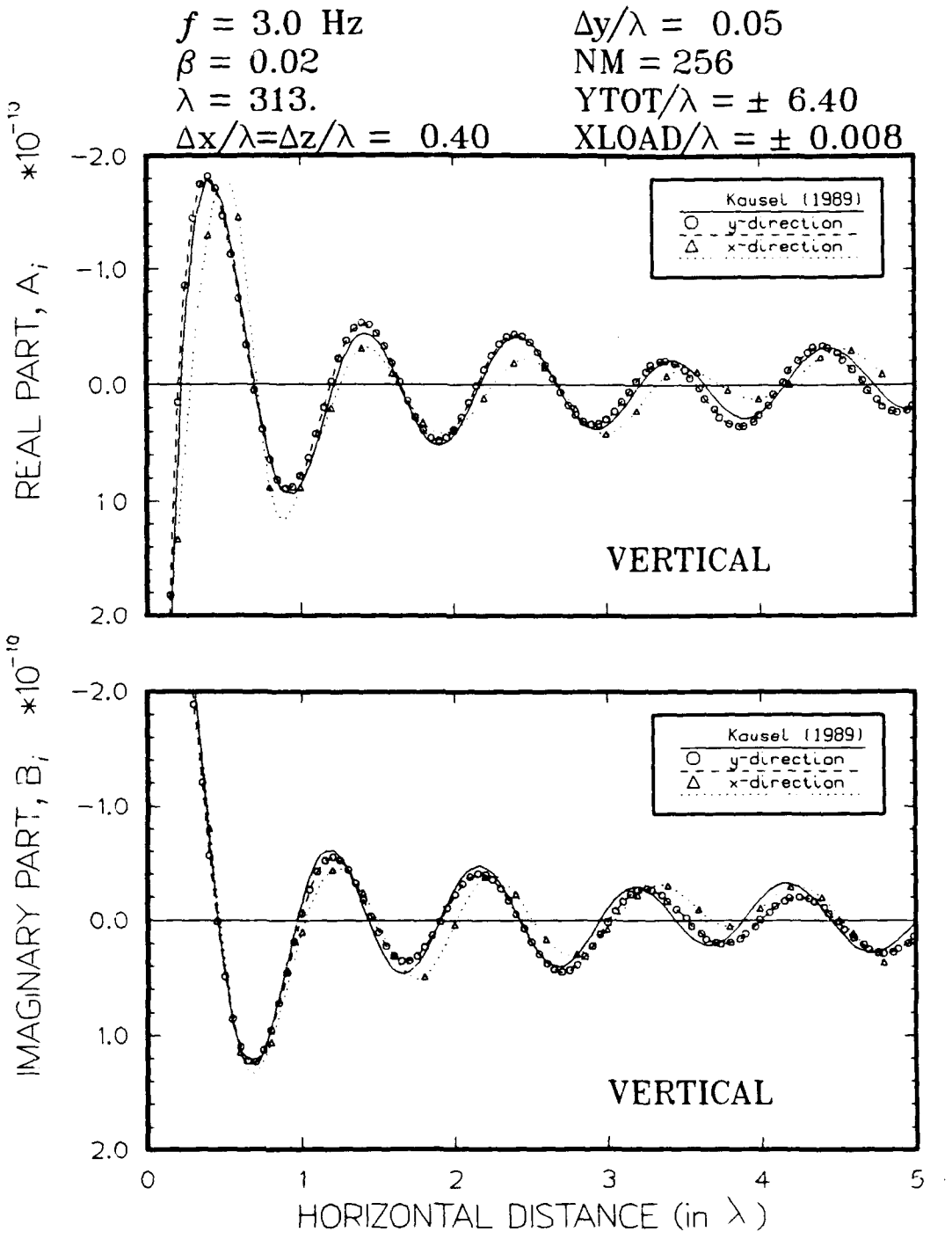


Figure 42. Vertical displacements discretized with 160 elements and Green's function solution

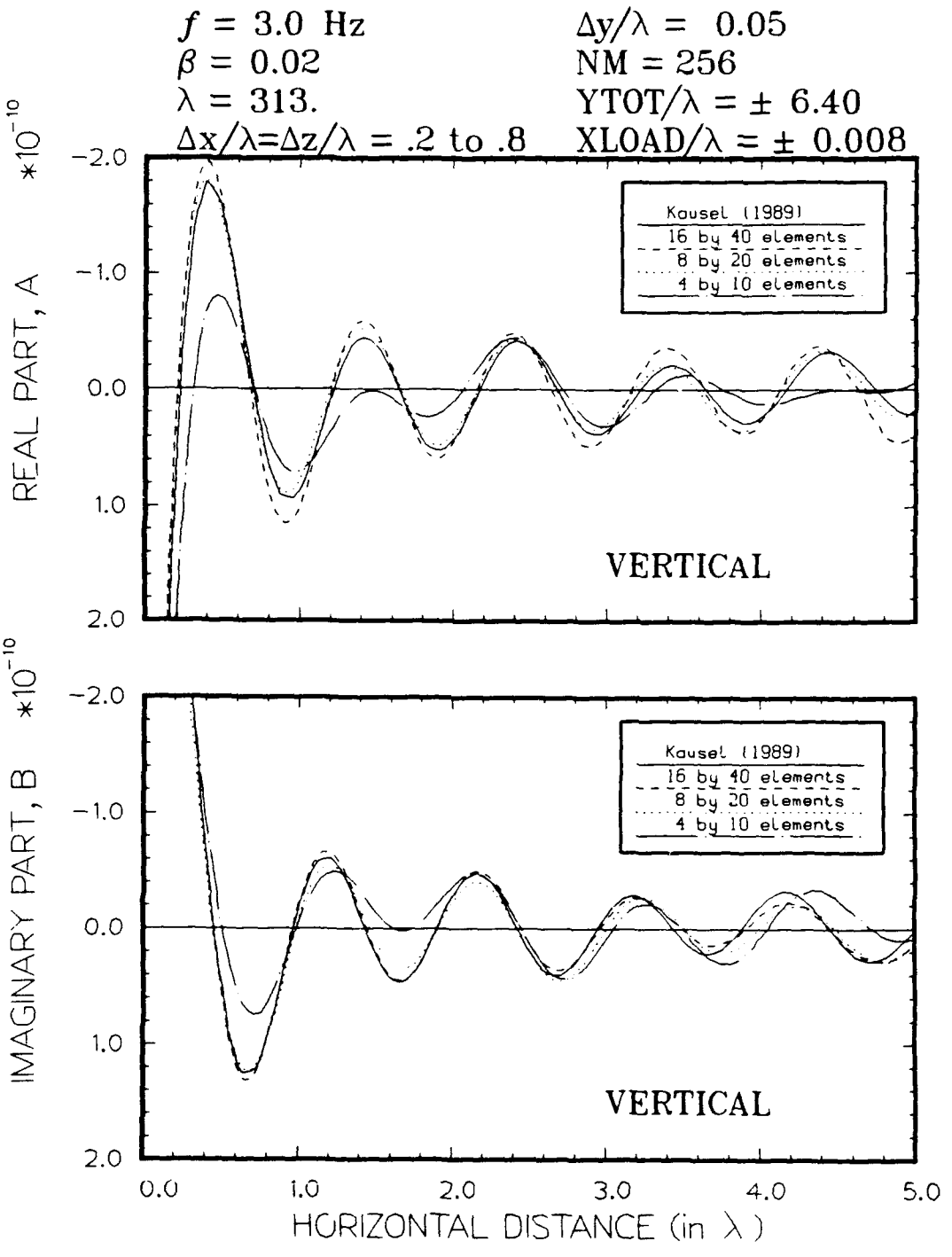


Figure 43. Comparison of vertical displacements showing effect of size of finite elements with Green's function solution

Effect of Width of Load

102. The load width in the x- and y-directions, XLOAD, ranges from a point load to $\pm$ 0.064λ (80 by 80 in total plan dimensions at 3 Hz). For all practical purposes at these distances and depths, these loads are essentially point loads. The dynamic displacements for each condition are presented in Figures 44 through 47 along with the Green's function solution. The results for the variation in the y-direction for the different load widths are compared in Figure 48.

103. Little noticeable effect is evident as XLOAD is varied over the specified range. The results for the largest width considered ($\pm$ 0.064λ) are slightly different from the other three sets and the Green's function solutions shown in Figure 19. A threshold of load $\leq \pm$ 0.10λ appears to be reasonable to maintain good accuracy.

104. The small difference between the results for the point load and the smallest square load is somewhat surprising. Kang (1990) noticed a larger difference and researchers have recognized the difficulty in calculating an accurate distribution of displacements from a point load using the finite element procedure without a refined mesh in the vicinity of the load.

Computational Effort

105. The amount of time necessary to run the program with different system parameters was reviewed. The two parameters considered to have the greatest effect are the number of FFT points, NM, and the number of degrees of freedom, dof. (Recall that the equations are solved for only half of the NM and the results mirrored prior to the inverse Fourier transform.) Comparisons of user CPU (central processing unit) times versus NM are shown in Figure 49. The three finite element meshes described earlier were used to provide a range in degrees of freedom. The solution times are of the same order as NM (linear relationship) for a fixed number of dof. The slopes of these lines range from 1.3 to 26. Using the relationship representing 6099 dof as an example, the increase in time to raise the total NM from 64 to 128 is 64 x 26 = 1664 sec. Comparisons of user CPU times versus dof for various NM are shown in Figure 50. The relationship is slightly non-linear for a fixed NM; the exponent of dof is about 1.12 and increases slightly as NM increases.

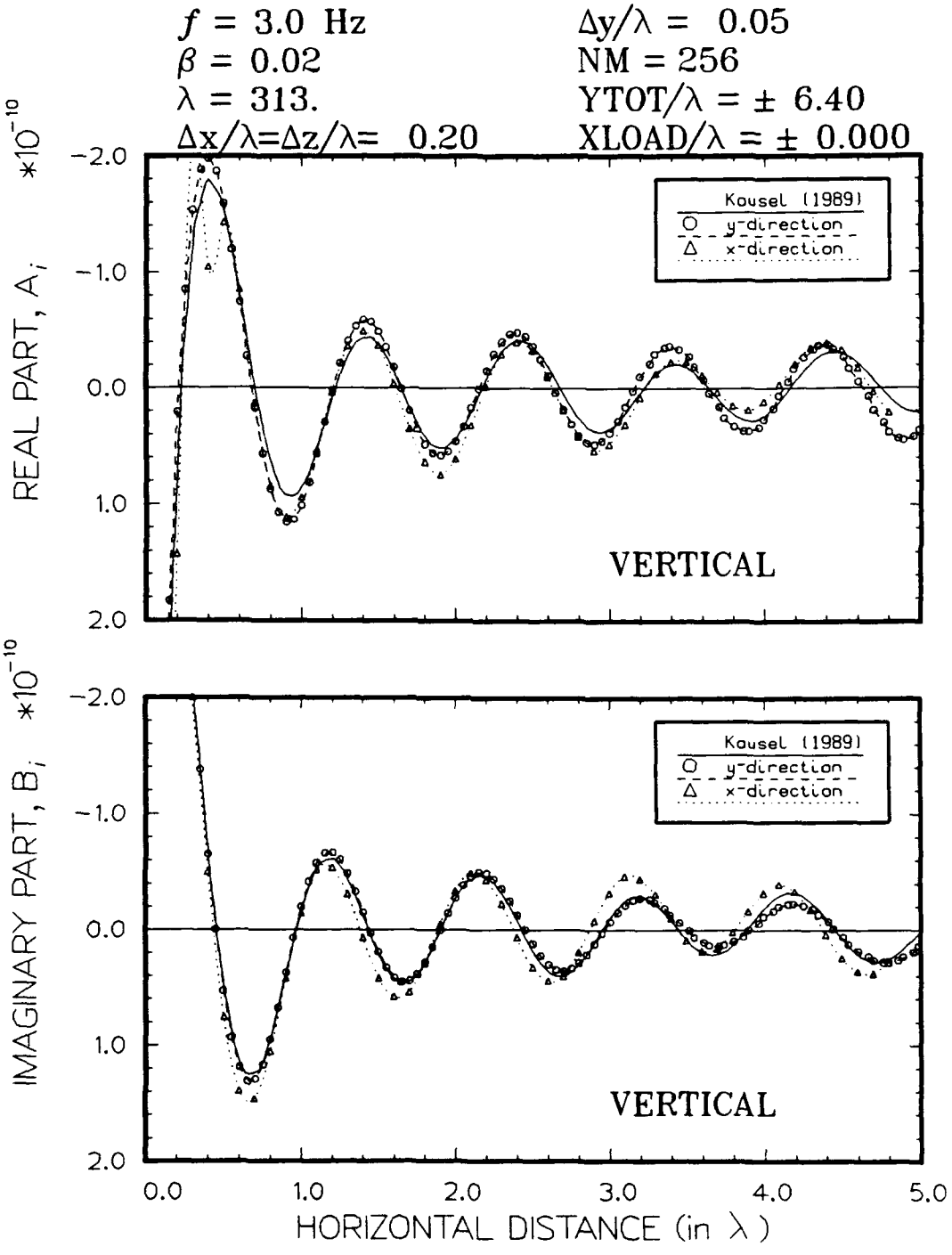


Figure 44. Vertical displacements subjected to a point load and Green's function solution

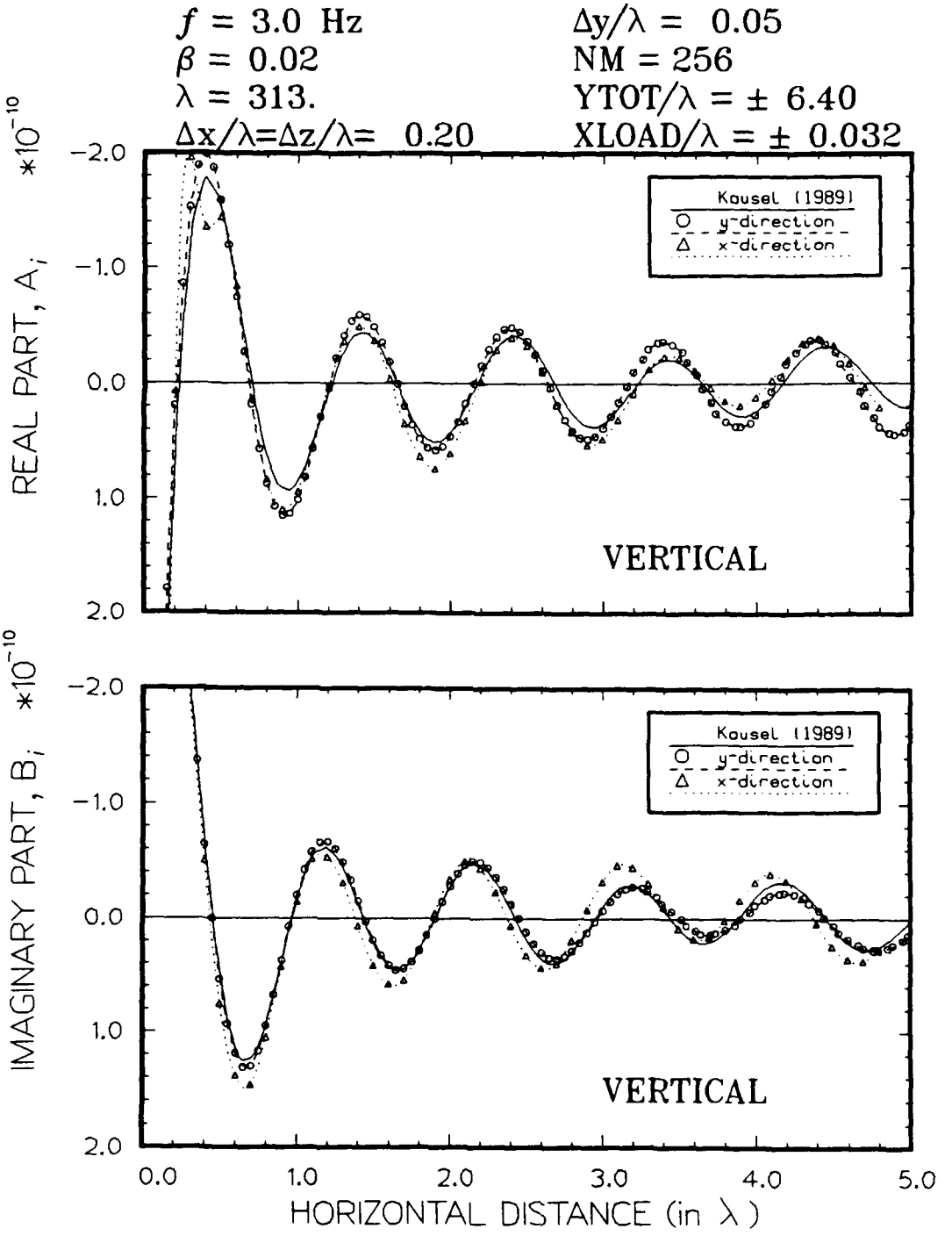


Figure 45. Vertical displacements subjected to a 0.06λ-square load and Green's function solution

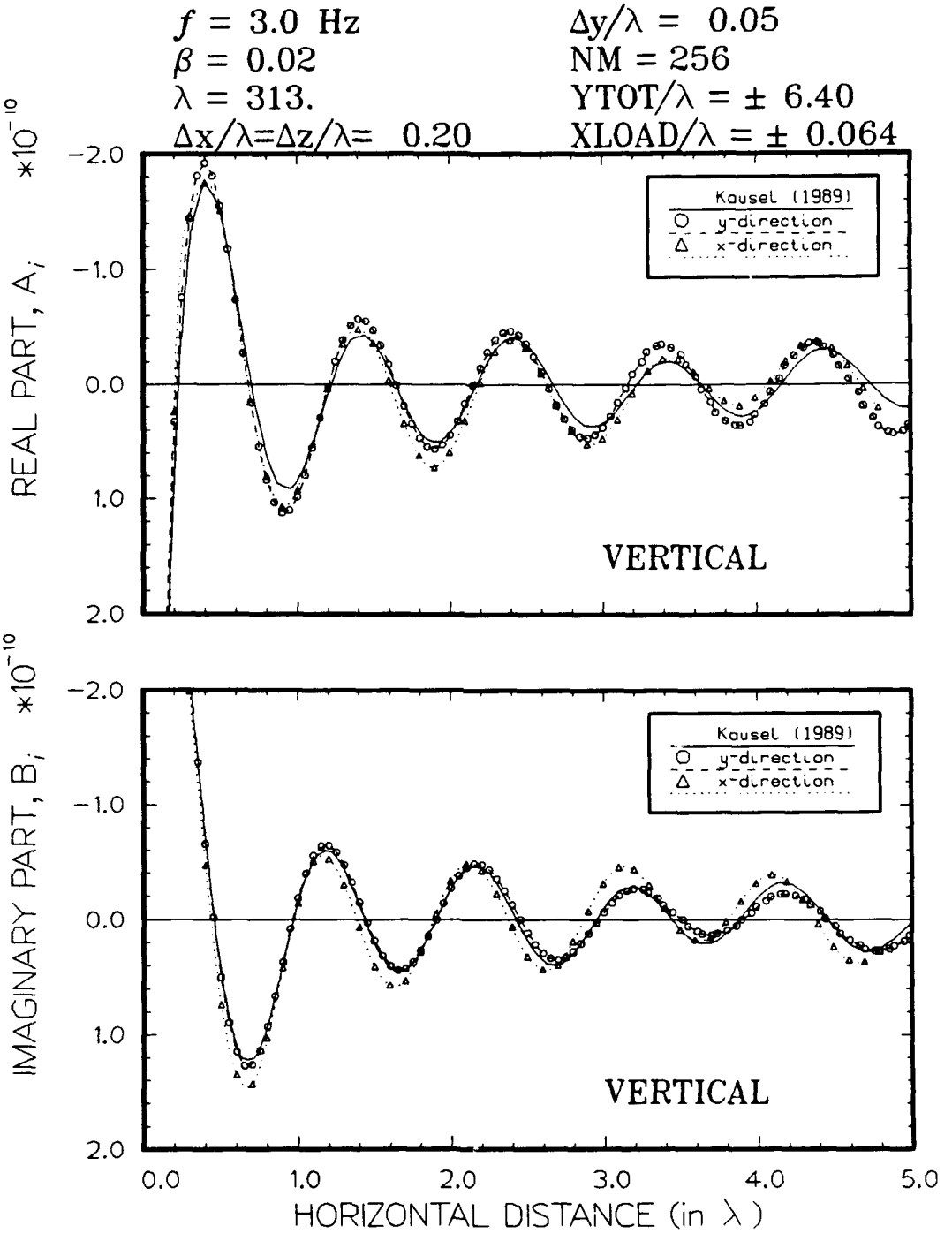


Figure 46. Vertical displacements subjected to a 0.13λ-square load and Green's function solution

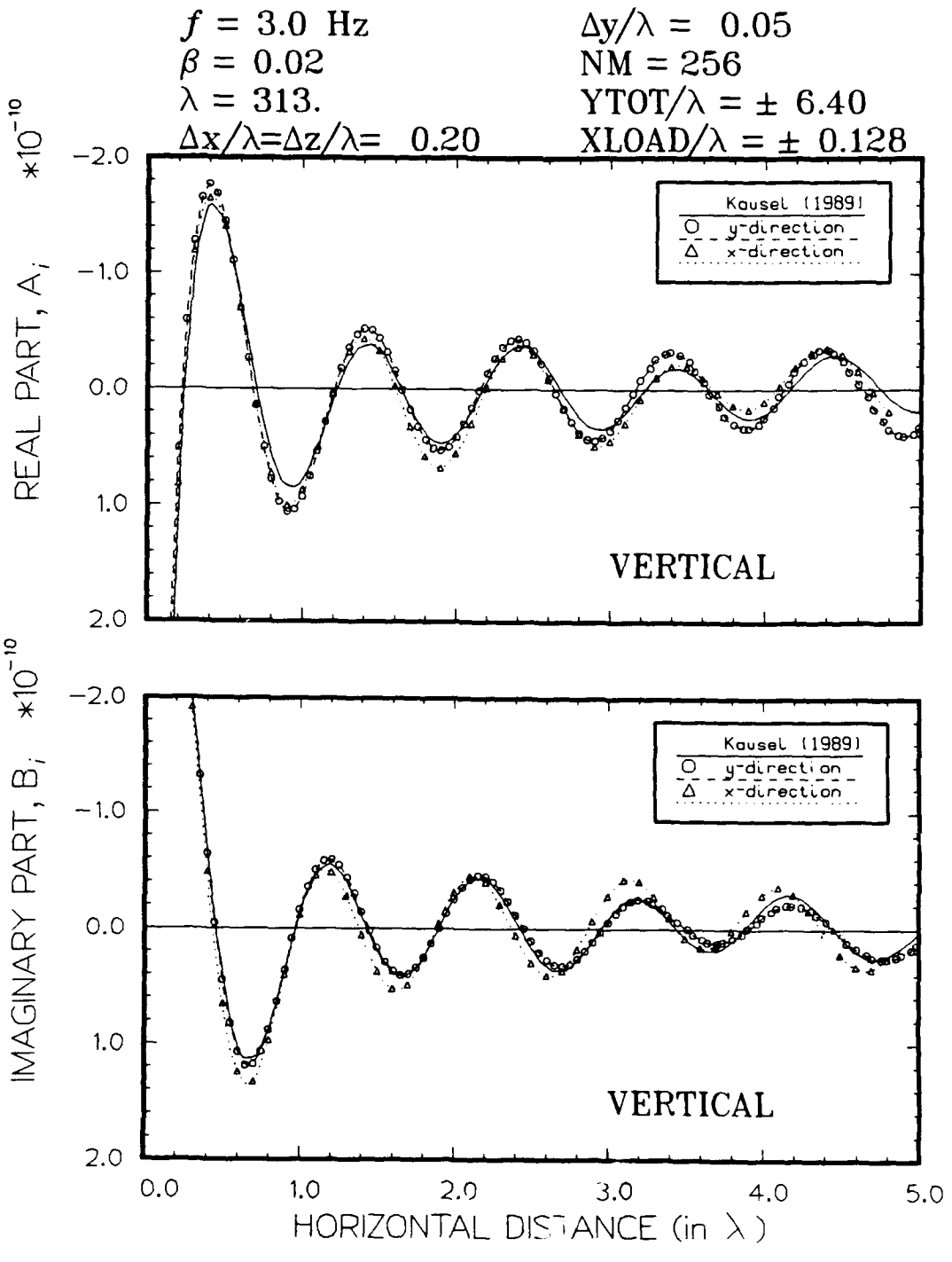


Figure 47. Vertical displacements subjected to a 0.26λ-square load and Green's function solution

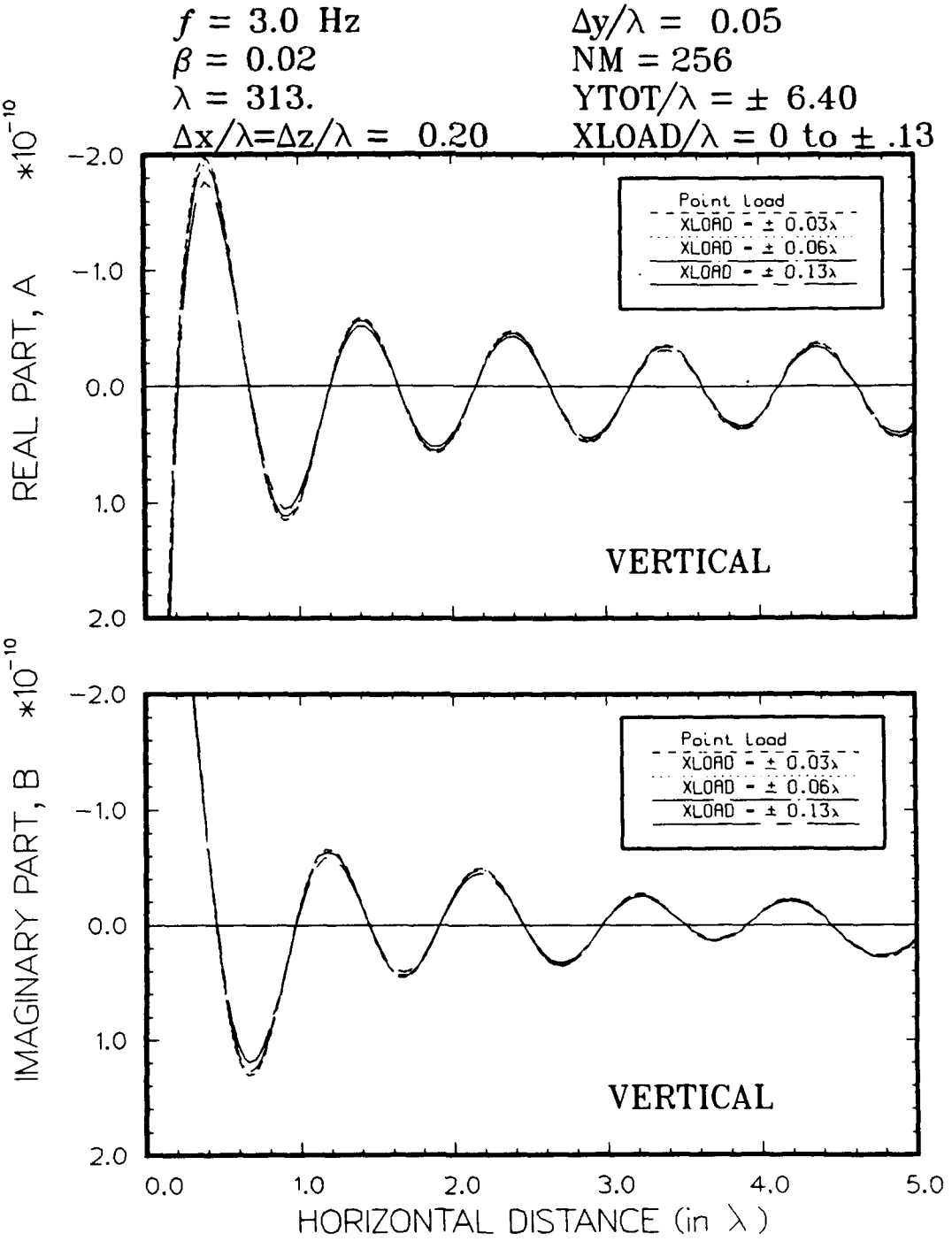


Figure 48. Comparison of vertical displacements showing effect of varying load width

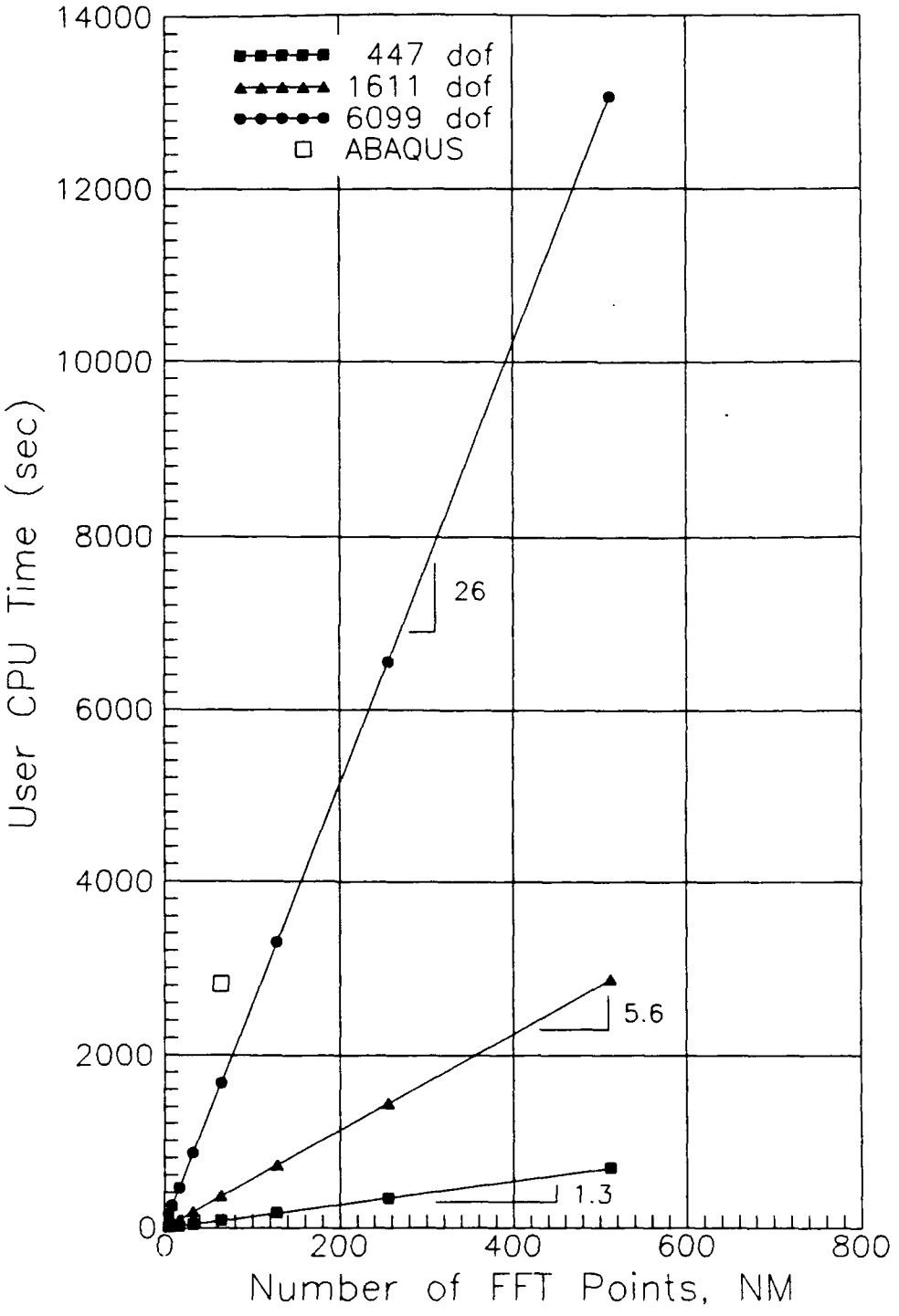


Figure 49. Comparison of CPU execution times on CRAY Y-MP versus number of FFT points

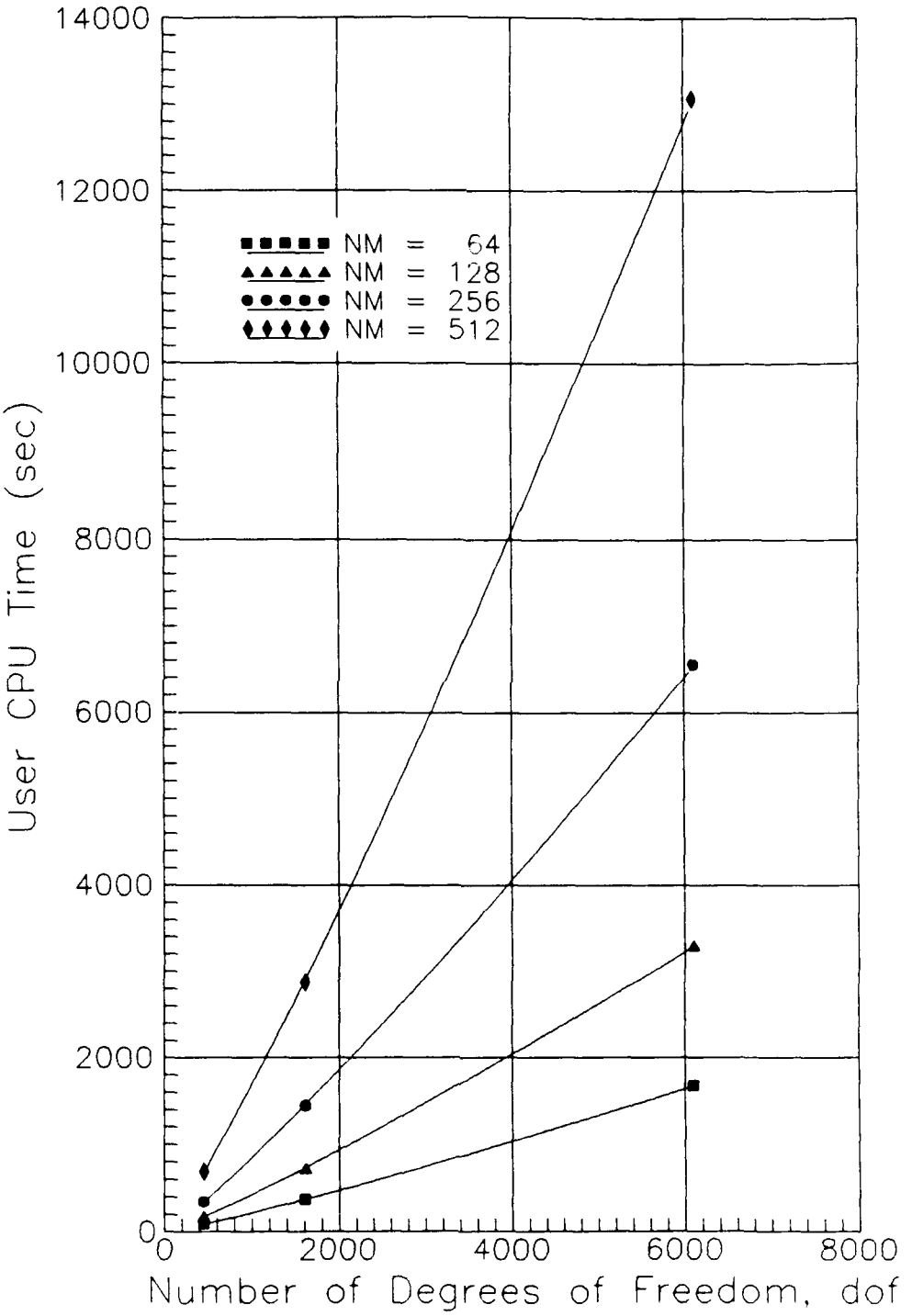


Figure 50. Comparison of CPU execution times on CRAY Y-MP versus number of degrees of freedom

106. The amount of time saved in using the present formulation over a conventional 3-D finite element formulation was estimated by solving the problem for Model 1 using the commercial software package *ABAQUS*. Two planes of symmetry were used such that only a 3-D quarter space was required to be discretized. A total discretized space of 8λ by 3λ in plan by 3λ deep was used and the element size was equal to that used in the 8 by 20 mesh ($\Delta x = \Delta y = \Delta z = 0.40\lambda$). A 3-D isoparametric element with 20 nodes (quadratic interpolation functions in all three directions) was selected. The extent and accuracy of discretization in the y-direction is roughly equivalent to NM = 64 and $\Delta y = 0.10\lambda$ which were used with *vib3*. Free end conditions were used for non-symmetric boundaries. The calculated results were not of particular interest. The user CPU time required by *ABAQUS* to solve for dynamic displacements was about 2820 sec compared to 370 sec using *vib3*. A comparison of times is shown in Figure 49. Moreover, almost 8 Mwords of memory were required to solve the problem using *ABAQUS* whereas about 3.6 Mwords were used by *vib3*.

Conclusions

107. The proposed formulation as implemented in the computer code *vib3* has been found to be sensitive to the following system parameters: the spatial increments of discretization in the y- and x-directions (Δz always set equal to Δx) and the total length of discretized space in the y-direction. However, the solutions in the x-direction are independent of Δy. The results are also moderately sensitive to reasonable ranges of Poisson's ratio and damping ratio. The solutions are not very sensitive to the width of the load using reasonable bounds for synthetic loads. The calculated displacements were again shown to be more accurate as damping ratio was increased from 2 to 5 percent.

108. The parametric analyses provided guidelines for the selection of system parameters to ensure acceptable performance. Thresholds generally confirming the results of findings by others are: $\Delta y \leq 0.05\lambda$ and $\Delta x \leq 0.30\lambda$. Thresholds established as a consequence of this work are: YTOT $\geq \pm 10\lambda$ and XLOAD $\leq \pm 0.10\lambda$.

PART VI: SUMMARY

109. A method to calculate dynamic displacements in 2-D geosystems produced by a harmonic point or rectangular load has been formulated and implemented in a two-dimensional finite element computer code and supporting pre- and post-processing programs that function on the US Army CRAY supercomputer at WES. This code has been validated with analytical solutions for the case of axi-symmetric geosystems subjected to static and dynamic loads. Parametric studies were performed to determine how the accuracy of the calculated displacements are affected by the various input parameters. All comparisons indicate that this method is a viable alternative to more time consuming 3-D numerical solution methods.

110. Pre-existing studies about Rayleigh wave propagation were reviewed to determine if alternative means are available to calculate dynamic displacements for the stated assumptions. None of the studies reviewed provided a solution to solve the stated problem. Some of the experimental studies provide insight into the propagation characteristics of surface waves around discontinuities and changes in ground slope. One study by Kausel (1981) was found to be appropriate to validate the code for the simplest case of axi-symmetric problems.

111. The formulation involves creating a 3-D dynamic stiffness matrix and then condensing the components into an equivalent 2-D dynamic stiffness matrix. The out-of-plane loads are represented by a Fourier expansion and applied as nodal forces. The solution to the system of equations is made for each spatial wavenumber and then the inverse Fourier transform produces the complex dynamic displacements.

112. The 2-D formulation implemented into the computer code, *vib3*, has been proven to provide accurate values of static and dynamic vertical displacements. Validation studies were performed for cases of static and dynamic loads generally using reasonable values of system parameters. The effects of static loads were examined in terms of displacement and stress field for cantilever beams in tension, compression, and torsion using the specialized 16-node, 3-D, finite element incorporated into a static 3-D finite element computer code. Calculated values were compared with closed-form elastic solutions. The displacements produced by static and dynamic point and square loads were examined for cases of a homogeneous medium and three

combinations of four-layered media using *vib3* and compared with Green's function solutions proposed by Kausel (1981).

113. The analysis of parameters necessary to the program indicates that once threshold values are met, the formulation is stable to variations in parameters defining the discretization, condensation, and Fourier expansion of the problem. These thresholds are: $\Delta y \leq 0.05\lambda$, YTOT $\geq \pm 10\lambda$, $\Delta x = \Delta z \leq 0.30\lambda$ (for quadratic interpolation), and XLOAD $\leq \pm 0.10\lambda$. Additional improvements may be realized by using even smaller values of Δx, Δy, and Δz. Displacements can be calculated about 8 times faster using the new formulation when compared to the 3-D finite element code *ABAQUS*.

114. The results of this study allow engineers to efficiently evaluate wave propagation for problems of vibration (e.g., effect of vehicular vibrations on sensitive equipment) and tunnel detection using a relatively simple representation of the system. Little knowledge about the mathematical formulation, or even the finite element method, are required.

PART VII: RECOMMENDATIONS

115. The next step in this research area should be the parametric numerical analysis of more complex systems and evaluation of dynamic displacements. This effort will be used to infer the best method and procedures for field measurements and develop a strategy for formulation of the inverse model. Following that analysis, calculated values should be compared with field measurements in real geotechnical systems One of the primary objectives of this comparison is to examine typical levels of signal-to-noise ratio and confirm that the desired signal is distinguishable within the desired range of distance. Other objectives are to measure the frequency band width of energy produced by a Vibroseis truck and evaluate its effect on measured signals and to determine the distance beyond which the assumption of plane geometry extending to infinity is no longer required.

REFERENCES

Alsop, L. E. 1966. " Transmission and Reflection of Love Waves at the Vertical Discontinuity," J. Geophys. Res., Vol 71, pp 3969-3984.

Apsel, R. J. 1979. "Dynamic Green's Function for Layered Media and Applications to Boundary Value Problems," PhD thesis, Univ. Calif., San Diego, CA.

Banerjee, P. K. and Mamoon, S. M. 1990. "A Fundamental Solution Due to a Periodic Point Force in the Interior of an Elastic Half-Space," Earthquake Engr. Struc. Dyn., Vol 19, pp 91-105.

Bard, P. Y. and Bouchon, M. 1980. "The Seismic Response of Sediment-Filled Valleys, Part 2: The Case of Incident P and SV Waves," Bull. Seism. Soc., Vol 70, No. 5, pp 1921-1941.

Barkan, D. D. 1962. Dynamics of Bases and Foundations, Sec. VIII-7, McGraw-Hill, New York, NY, (trans. from Russian).

Bathe, K. J. 1982. Finite Element Procedures in Engineering Analysis, Prentice-Hall, Inc., Englewood Cliffs, N.J.

deBremaecker, J. C. 1958. "Transmission and Reflection of Rayleigh Waves at Corners," Geophysics, Vol 23, No. 2, pp 253-266.

Clough, R. W. and Penzien, J. 1975. Dynamics of Structures, McGraw-Hill, New York, NY.

Cooley, J. W. and Tukey, J. W. 1965. "An Algorithm for Machine Calculation of Complex Fourier Series," Math Computation, Vol 19, pp 297-301.

Dolling, H. J. 1965. "Schwingungsisolierung von Bauwerken durch tiefe, auf geeignete Weise stabilisierte Schiltze," Xonderdruch aus VDI-Berichte 88, S. 3741.

Dolling, H. J. 1970. "Die Abschirmung von Erschutterungen durch Bodenschlitze," Die Bautechnick, pp 151-158.

Drake, L. A. 1972. "Rayleigh Waves at a Continental Boundary by the Finite Element Method," Bull. Seism. Soc., Vol 62, No. 5, pp 1259-1268.

Dravinski, M. 1980. "Scattering of Elastic Waves by an Alluvial Valley of Arbitrary Shape," Rpt. No. CE 80-06, Univ. South. Calif., Los Angeles, CA.

Dravinski, M. and Mossessian, T. K. 1987. "Scattering of Plane Harmonic P, SV, and Rayleigh Waves by Dipping Layers of Arbitrary Shape," Bull. Seism. Soc., Vol 77, No. 1, pp 212-235.

Dunkin, J. W. 1965. "Computation of Modal Solutions in Layered, Elastic Media at High Frequencies," Bull. Seism. Soc., Vol 55, No. 2, pp 335-358.

Eshraghi, H. and Dravinski, M. 1989a. "Transient Scattering of Elastic Waves by Dipping Layers of Arbitrary Shape, Part 2: Plane Strain Model," Earthquake Engr. Struc. Dyn., Vol 18, pp 417-434.

Eshraghi, H. and Dravinski, M. 1989b. "Scattering of Plane Harmonic SH, SV, P, and Rayleigh Waves by Non-Axisymmetric Three-Dimensional Canyons: A Wave Function Expansion Analysis," Earthquake Engr. Struc. Dyn., Vol 18, pp 983-998.

Eshraghi, H. and Dravinski, M. 1991. "Transient Scattering of Elastic Waves by Three-Dimensional Non-Axisymmetric Dipping Layers," Int'l J. Num. Meth. Engr., Vol 31, pp 1009-1026.

Fujii, K., Takeuchi, S., Okano, Y., and Nakano, M. 1984. "Rayleigh Wave Scatterings at Various Wedge Corners," Bull. Seism. Soc., Vol 74, No. 1, pp 41-60.

Fujii, K., Takimoto, T., Nakayama, Y., and Nakano, M. 1980. "Group of Rayleigh Waves Transmitted Around Trench on Surface of Elastic Half Space (II), J. Seis. Soc. Japan, Vol 33, pp 11-22.

Fuyuki, M. and Matsumoto, Y. 1980. "Finite Difference Analysis of Rayleigh Wave Scattering at a Trench," Bull. Seism. Soc., Vol 70, pp 2051-2069.

Fuyuki, M. and Nakano, M. 1984. "Finite Difference Analysis of Rayleigh Wave Transmission Past an Upward Step Change," Bull. Seism. Soc., Vol 74, No. 3, pp 893-911.

Gautesen, A. K. 1985. "Scattering of a Rayleigh Wave by an Elastic Quarter Space," Trans. ASME, Vol 52, pp 664-668.

Gray, H. 1936. "Stress Distribution in Elastic Solids," Proc., Int'l Conf. Soil Mech., Vol 2, Cambridge, MA, pp 157-168.

Gutdeutsch, R. 1969. "On Rayleigh Waves in a Wedge With Free Boundaries," Bull. Seism. Soc., Vol 59, No. 4, pp 1645-1652.

Hanazato, T., Ugai, K., Mori, M., and Sakaguchi, R. 1991. "Three-Dimensional Analysis of Traffic-Induced Ground Vibrations, J. Geotech. Engr., Vol 117, No. 8, pp 1133-1151.

Hardin, B. O. and Drnevich, V. P. 1972. "Shear Modulus and Damping in Soils: Measurement and Parameter Effects," J. Geotech. Engr., Vol 98, No. 6, pp 603-624.

Harkrider, D. G. 1964. "Surface Waves in Multilayered Elastic Media: I. Rayleigh and Love Waves From Buried Sources in a Multilayered Elastic Half-Space," Bull. Seism. Soc., Vol 54, No. 2, pp 627-679.

Haskell, N. A. 1953. "The Dispersion of Surface Waves in Multilayered Media," Bull. Seism. Soc., Vol 43, No. 1, pp 17-34.

Herrera, I. 1964. "A Perturbation Method for Elastic Wave Propagation," J. Geophys. Res., Vol 69, No. 18, pp 3845-3851.

Hudson, J. A. and Knopoff, L. 1964. "Transmission and Reflection of Surface Waves at a Corner: 2. Rayleigh Waves (Theoretical)," J. Geophys. Res., Vol 69, No. 2, pp 281-289.

Hudson, J. A. and Knopoff, L. 1967. "Scattered Surface Waves from a Surface Obstacle," Geoph. J. Res. Ast. Soc., Vol 13, pp 441-458.

Irons, B. M. 1970. "A Frontal Solution Program," Int. J. Num. Meth. Engr., Vol 2, pp 5-32.

Its, E. N. and Yanovskaya, T. B. 1977. "Reflection and Transmission of Rayleigh Waves at a Vertical Interface," in: Pattern Recognition and Spectral Analysis in Seismology, (Computational Seismology, 10), Moscow, pp 214-222, (in Russian).

Its, E. N. and Yanovskaya, T. B. 1979. "Reflection and Transmission of Surface Waves at Oblique Incidence on a Vertical Discontinuity," in: Theory and Analysis of Seismological Observations, (Computational Seismology, 12), Moscow, pp 86-92, (in Russian).

Its, E. N. and Yanovskaya, T. B. 1983. "Reflection and Transmission of Surface Waves at an Inclined Discontinuity," in: Earthquake Prediction and the Study of Earth Structure, (Computational Seismology, 15), Moscow, pp 87-92, (in Russian).

Johnston, D. H., Toksöz, M. N., and Timur, A. 1979. "Attenuation of Seismic Waves in Dry and Saturated Rocks: II. Mechanisms," Geophysics, Vol 44, pp 691-711.

Kane, J. and Spence, J. 1963. "Rayleigh Wave Transmission on Elastic Wedge," Geophysics, Vol 28, No. 5, Part I, pp 715-723.

Kang, Y. V. 1990. "Effect of Finite Width on Dynamic Deflections of Pavements," PhD dissertation, Univ. Texas, Austin, TX.

Kato, Y. and Takagi, A. 1956. "Wave Propagation in the Step Shaped Structure and on a Cliff," Rpt., Tohaku Univ., Vol 8, pp 74-85.

Kausel, E. 1981. "Explicit Solution for the Green Functions for Dynamic Loads in Layered Media," MIT Research Report R81-13, Cambridge, MA.

Kausel, E. 1989. "PUNCH: Program for the Dynamic Analysis of Layered Soils," ver. 3.0, Massachusetts Institute of Technology, Cambridge, MA.

Kausel, E. and Roesset, J. 1981. " Stiffness Matrices for Layered Soils," Bull. Seism. Soc., Vol 71, No. 6.

Kawase, H. 1988. "Time-Domain Response of a Semi-Circular Canyon for Incident SV, P, and Rayleigh Waves Calculated by the Discrete Wavenumber Boundary Element Method," Bull. Seism. Soc., Vol 78, No. 4, 1415-1437.

Khair, K. R., Datta, S. K., and Shah, A. H. 1991. "Amplification of Obliquely Incident Seismic Waves by Cylindrical Alluvial Valley of Arbitrary Cross-Sectional Shape, Part II: Incident SH and Rayleigh Waves," Bull. Seism. Soc., Vol 81, No. 2, pp 346-357.

Knopoff, L. 1969. "Elastic Wave Propagation in a Wedge," Wave Propagation in Solids, ed. Miklowitz, ASME, pp 3-42.

Knopoff, L. and Gangi, A. F. 1960. "Transmission and Reflection of Rayleigh Waves By Wedges," Geophysics, Vol 25, No. 6, pp 1203-1214.

Kuo, J. and Nafe, J. 1962. "Periodic Equation of Rayleigh Waves in a Layer Overlying a Half Space with Sinusoidal Interface," Bull. Seis. Soc., Vol 52, pp 807-822.

Kuo J. T. and Thompson, G. A. 1963. "Model Studies on the Effect of a Sloping Interface on Rayleigh Waves," J. Geophys. Res., Vol 68, No. 2, pp 6187-6197.

Lai, J. Y. and Booker, J. R. 1991. "Application of Discrete Fourier Series to the Finite Element Stress Analysis of Axi-Symmetric Solids," Int'l J., Num. Meth. Engr., Vol 31, pp 619-647.

Lee, V. W. 1978. "Displacements Near a Three-Dimensional Hemispherical Canyon Subject to Incident Plane Waves," Rpt. No. CE 78-16, Univ. South. Calif, Los Angeles, CA.

Lee, J. J. and Langston, C. A. 1983. "Wave Propagation in a Three-Dimensional Circular Basin," Bull. Seis. Soc., Vol 73, No. 6, pp 1637-1653.

Lee, V. W. 1984. "Three-Dimensional Diffraction of Plane P, SV, & SH Waves by a Hemispherical Alluvial Valley," Soil Dyn. Earthquake Engr., Vol 3, No. 3, pp 133-144.

Li, Z. L. and Achenbach, J. D. 1991. "Reflection and Transmission of Rayleigh Surface Waves by a Material Interphase," Trans., ASME, Vol 58, pp 688-694.

Lin, H-T. and Tassoulas, J. L. 1987. "Discrete Green Functions for Layered Strata," Int'l J. Num. Meth. Engr., Vol 24, pp 1645-1658.

Lopez-Soto, L. A. 1967. "Propagacion de Ondas de Rayleigh en Cuñas Elasticas," Pub. 47, Dept. Geophysics & Geodesy, Univ. Chile, Santiago.

Love, A. E. H. 1944. A Treatise on the Mathematical Theory of Elasticity, 4th ed., Dover, New York, NY.

Lutikov, A. I. 1979. "The Effect of an Inclined Discontinuity on the Reflection and Transmission of Rayleigh Waves," Izv. AN SSSR, Fizika Zemli (Solid Earth), No. 10, pp 58-66.

Lysmer, J. and Duncan, J. M. (ed.) 1969. "Stresses and Deflections in Foundations and Pavements," 4th ed., Dept. Civil Engr., Univ. Calif., Berkeley, CA.

Mal, A. K. and Knopoff, L. 1965. "Transmission of Rayleigh Waves Past a Step Change in Elevation," Bull. Seism. Soc., Vol 55, No. 2, pp 319-334.

Mal, A. K. and Knopoff, L. 1966. "Transmission of Rayleigh Waves at a Corner," Bull. Seism. Soc., Vol 56, No. 2, pp 455-466.

Malischewski, P. 1974. "The Influence of Curved Discontinuities on the Propagation of Surface Waves," Gerlands Beitr. Geophys., Vol 83, pp 355-362.

Malischewski, P. 1976. "Surface Waves in Media Having Lateral Inhomogeneities," Pure & Applied Geophys., Vol 114, pp 833-843.

McGarr, A. and Alsop, L. E. 1967. "Transmission and Reflection of Rayleigh Waves at Vertical Boundaries," J. Geophys. Res., Vol 72, No. 8, pp 2169-2180.

Milder, D. M. 1991. "An Improved Formalism for Wave Scattering From Rough Surfaces," J. Acoust. Soc. Am., Vol 89, No. 2, pp 529-541.

Mindlin, R. D. 1936. "Force at a Point in the Interior of a Semi-Infinite Solid," J. Appl. Phys., Vol 7, No. 5, pp 195-202.

Mooney, H. M. 1974. "Some Numerical Solutions for Lamb's Problem," Bull. Seism. Soc., Vol 64, No. 2, pp 473-491.

Nazarian, S. and Stokoe, K. H. II, 1985a. "In Situ Determination of Elastic Moduli of Pavement Systems by Spectral-Analysis-of-Surface-Waves Method (Practical Aspects)," Research Report No. 368-1F, Center Transportation Research, Austin, TX.

Nazarian, S. and Stokoe, K. H. II, 1985b. "In Situ Determination of Elastic Moduli of Pavement Systems by Spectral-Analysis-of-Surface-Waves Method (Theoretical Aspects)," Research Report No. 437-1, Center Transportation Research, Austin, TX.

Ohtsuki, A. and Yamahara, H. 1984. "Effect of Topography and Subsurface Inhomogeneity on Seismic SV Waves and Rayleigh Waves," Proc., 8th World Earthquake Engr., Vol II, San Francisco, CA, pp 655-662.

Oliver, J., Press, F., and Ewing, M. 1954. "Two-Dimensional Model Seismology," Geophysics, Vol 19, pp 202-219.

Pekeris, C. L. 1955. "The Seismic Buried Pulse," Proc., Nat. Acad. Sci., Vol 41, pp 629-639.

Pinney, E. 1954. "Surface Motion Due to a Point Source in a Semi-Infinite Elastic Medium," Bull. Seism. Soc., Vol 44, pp 571-596.

Pestel, E. C. and Leckie, F. A. 1963. Matrix Methods in Elastomechanics, McGraw-Hill, New York, NY.

Pilant, W. L., Knopoff, L., and Schwab, F. 1964. "Transmission and Reflection of Surface Waves at a Corner: 3. Rayleigh Waves (Experimental)," J. Geophys. Res., Vol 69, No. 2, pp 291-297.

Poulos, H. G. and Davis, E. H. 1972. Elastic Solutions for Soil and Rock Mechanics, John Wiley and Sons, New York, NY.

Rix, G. J. and Leipski, E. A. 1991. "Accuracy and Resolution of Surface Wave Inversion," Recent Advances in Instrumentation, Data Acquisition, and Testing in Soil Dynamics, ASCE, Geot. Pub. No. 29, ed. Bhatia and Blaney, pp 17-32.

Runesson, K. and Booker, J. R. 1982. "Efficient Finite Element Analysis of 3D Consolidation," Numerical Methods Geomechanics, Edmonton 1982, ed. Eisenstein, Balkema Press, Rotterdam, pp 359-364.

Runesson, K. and Booker, J. R. 1983. "Finite Element Analysis of Elastic-Plastic Layered Soil Using Discrete Fourier Series Expansion," Int'l J. Num. Meth. Engr., Vol 19, pp 473-478.

Sanchez-Sesma, F. J. 1983. "Diffraction of Elastic Waves by Three-Dimensional Surface Irregularities," Bull. Seis. Soc., Vol 73, No. 6, pp 1621-1636.

Sanchez-Sesma, F. J., Bravo, J. A., and Herrera, I. 1985. "Surface Motion of Topographical Irregularities for Incident P, SV, and Rayleigh Waves," Bull. Seism. Soc., Vol 75, No. 1, pp 263-269.

Sanchez-Sesma, F. J., Chavez-Perez, S., and Aviles, J. 1984. "Scattering of Elastic Waves by Three-Dimensional Topographies," Proc., 8th World Earthquake Engr., Vol II, San Francisco, CA, pp 639-646.

Sanchez-Sesma, F. J., Perez-Rocha L. E., and Chavez-Perez, S. 1985. "Diffraction of Elastic Waves by Three-Dimensional Surface Irregularities, Part II," Bull. Seism. Soc., Vol 79, pp 101-112.

Scheidl, W. and Ziegler, F. 1977. "Interaction of a Pulsed Rayleigh Surface Wave and a Rigid Cylindrical Inclusion," Modern Problems in Elastic Wave Propagation, ed. Miklowitz and Achenbach, Wiley-Interscience, New York, NY, pp 145-169.

Segol, G., Lee, P. C. Y., and Abel, J. F. 1978. "Amplitude Reduction of Surface Waves by Trenches," J. Engr. Mech. Div., ASCE, Vol 104, No. EM3, pp 621-641.

Stroud, A. H. and Secrest, D. 1966. "Gaussian Quadrature Formulas," Prentice-Hall, Englewood Cliffs, N.J.

Thomson, W. T. 1950. "Transmission of Elastic Waves Through a Stratified Solid Medium," J. Sound Vib., Vol 2, No. 3, pp 210-226.

Timoshenko, S. P. and Goodier, J. N. 1970. Theory of Elasticity, McGraw-Hill, New York, NY.

Toksöz, M. N., Johnston, D. H., and Timur, A. 1979. "Attenuation of Seismic Waves in Dry and Saturated Rocks: II. Laboratory Measurements," Geophysics, Vol 44, pp 681-690.

Uberall, H. 1977. "Modal and Surface Wave Resonances in Acoustic-Wave Scattering from Elastic Objects and in Elastic-Wave Scattering from Cavities," Modern Problems in Elastic Wave Propagation, ed. Miklowitz and Achenbach, Wiley-Interscience, New York, NY, pp 239-263.

Udaka, T. and Lysmer, J. 1973. "Supplement to Computer Program SHAKE," Univ. Calif., Berkeley, CA.

Vardoulakis, I. and Vrettos, C. 1988. "Dispersion-Law of Rayleigh-Type Waves in a Compressible Gibson Half-Space," Int'l. J. Num. Anal. Meth. Geomech., Vol 12, pp 639-655.

Viktorov, A. 1958. "The Effects of Surface Defects on the Propagation of Rayleigh Waves," Acoustical Institute, Academy of Sciences, USSR, pp 304-306.

Waas, G. 1972a. "Earth Vibration Effects and Abatement for Military Facilities, Report 3: Analysis Method for Footing Vibrations Through Layered Media," Technical Report S-71-14, Chptr 10, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Waas, G. 1972b. "Linear Two Dimensional Analysis of Soil Dynamics Problems in Semi-Infinite Layered Media," Phd thesis, Univ. Calif., Berkeley, CA.

Wong, H. L. 1982. "Effect of Surface Topography on the Diffraction of P, SV, and Rayleigh Waves," Bull. Seism. Soc., Vol 72, No. 4, pp 1167-1183.

Winnicki, L. A. and Zienkiewicz, O. C. 1979. "Plastic (or Visco-Plastic) Behaviour of Axisymmetric Bodies Subjected to Non-Symmetric Loading -- Semi-Analytical Finite Element Solutions," Int'l J., Num. Meth. Engr., Vol 14, pp 1399-1412.

Wolf, J. P. 1985. Dynamic Soil-Structure Interaction, Prentice-Hall, Inc., Englewood Cliffs, NJ, pp 15-16.

Woods, R. D. 1968. "Screening of Surface Waves in Soils," J., Soil Mech. Found. Div., ASCE, Vol 94, No. SM4, pp 951-979.

Yanovskaya, T. B. 1989. Seismic Surface Waves in a Laterally Inhomogeneous Earth, ed. V. I. Keilis-Borok, Kluwer Academic, Boston, MA.

Zienkiewicz, O. C. and Taylor, R. L. 1989. The Finite Element Method, Vol 1: Basic Formulation and Linear Problems, McGraw-Hill, London, 4th ed.

APPENDIX A:

STATIC LOADS ON SEMI-INFINITE MEDIA

A1. Closed-form solutions for (static) vertical and horizontal loads acting on the surface of semi-infinite, elastic media are manipulated to show the importance of properly formulating a two-dimensional ("planar") problem subjected to three-dimensional loads. The explicit solutions by Boussinesq and Cerutti form the basis for this demonstration. The vertical point load represents a 3-D load and the line load represents the equivalent 2-D load. The conventions used for definitions of variables are shown in Figure A-1. Solutions presented by Boussinesq and Cerutti for point loads were evaluated on the plane y = 0 (in-plane solution). Unit-less dimensions are used throughout; consistent units should be used when involving the equations presented. (The form of units is shown in square brackets where F = force, L = length, and T = time.)

Vertical loads

A2. Boussinesq published explicit solutions for the determination of stresses and displacements in semi-infinite media caused by a vertical point load acting at the surface. The medium is assumed to be homogeneous, isotropic, and linear elastic. The closed-form Boussinesq solutions, as reported by Gray (1936), for vertical stress, σ_v, horizontal stress, σ_h, and shear stress on a vertical plane, τ_{xz}, all in the form $[F/L^2]$ produced by a vertical point load are:

$$\sigma_z^{pt} = \frac{3Pz^3}{2\pi R^5} \tag{A1}$$

$$\sigma_x^{pt} = \frac{-P}{2\pi R^5}\left[\frac{-3x^2z}{R^3} + \frac{(1-2\nu)R}{R+z}\right] \tag{A2}$$

$$\tau_{xz}^{pt} = \frac{3Pxz^2}{2\pi R^5} \tag{A3}$$

where

P = magnitude of vertical point load [F]
ν = Poisson's ratio [-]

Notice that σ_z^{pt} and τ_{xz}^{pt} are independent of material properties; only σ_x^{pt} (Equation 2) is a function of a material property, ν.

A3. Equations 1 through 3 can be integrated to obtain explicit solutions caused by an infinite vertical line load as reported by Gray (1936). The closed-form solutions for a vertical line load acting on the surface are:

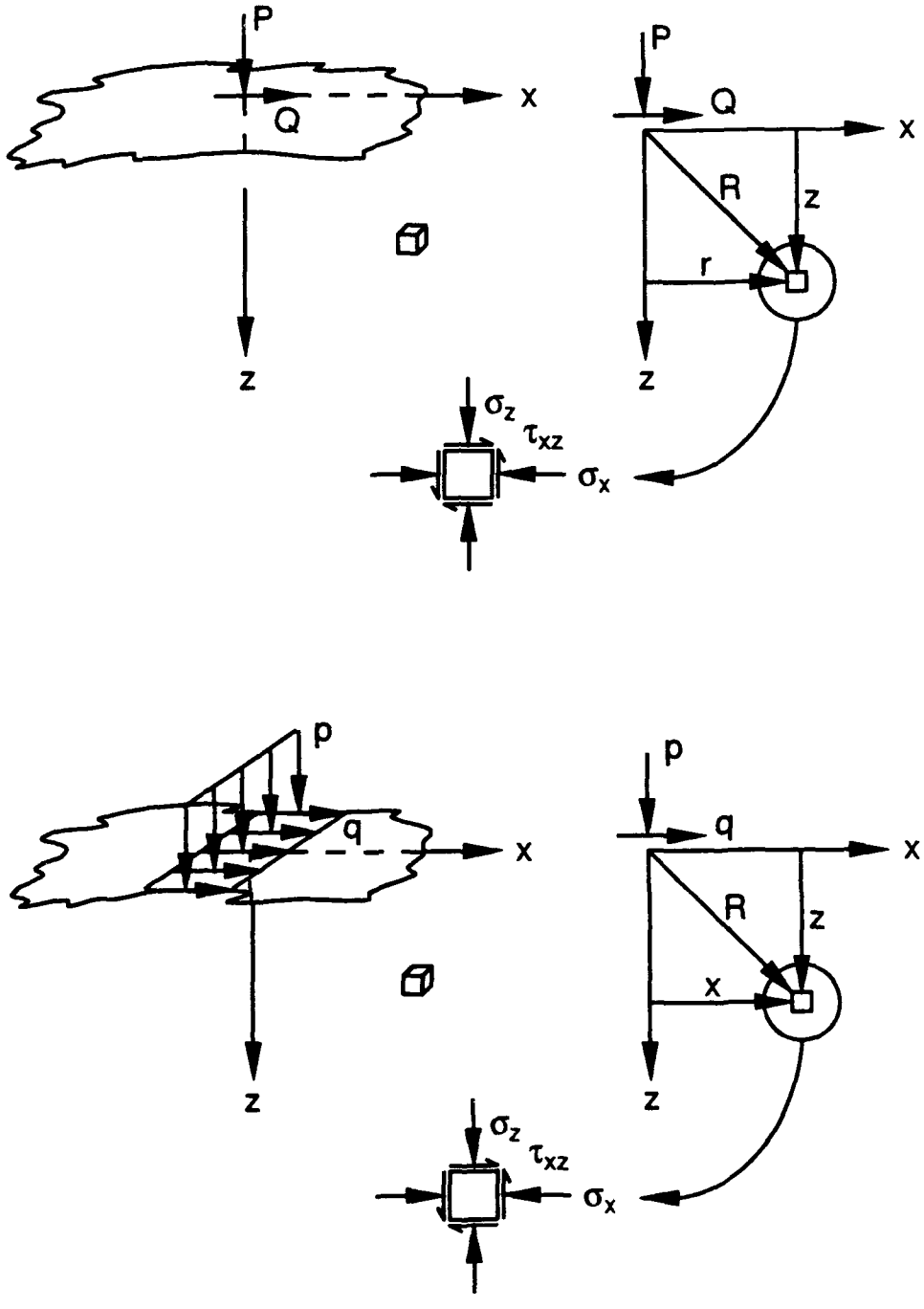


Figure A-1. Definition of variables for point and line load problems

$$\sigma_z^{ln} = \frac{2pz^3}{\pi R^4} \tag{A4}$$

$$\sigma_x^{ln} = \frac{2px^2z}{\pi R^4} \tag{A5}$$

$$\tau_{xz}^{ln} = \frac{2pxz^2}{\pi R^4} \tag{A6}$$

where

p = magnitude of vertical line load [F/L]

All three equations presented for line loads are independent of material properties. Solutions for line loads of finite length can be obtained in the form of tabulated influence factors by Lysmer and Duncan (1969).

A4. The explicit solutions for vertical point and line loads can be readily compared by forming the ratios of the respective stresses. The ratios of stresses acting on a vertical plane for vertical point and lines loads are:

$$\frac{\sigma_z^{pt}}{\sigma_z^{ln}} = \frac{3}{4R}\frac{P}{p} \tag{A7}$$

$$\frac{\sigma_x^{pt}}{\sigma_x^{ln}} = \frac{R^3}{4x^2}\left[\frac{3x^2}{R^4} - \frac{(1-2\nu)}{z(R+z)}\right]\frac{P}{p} \tag{A8}$$

$$\frac{\tau_{xz}^{pt}}{\tau_{xz}^{ln}} = \frac{3}{4R}\frac{P}{p} \tag{A9}$$

Notice that the ratios formed in Equations A7 and A9 are equivalent.

A5. The functions defined for the ratios shown in Equations A7 (or A9) and A8 are plotted as three-dimensional surfaces in Figures A-2 and A-3 and Figures A-4 and A-5, respectively, for different combinations of load to facilitate comparison. These surfaces are also represented with two-dimensional contours subimposed with the surface.

A6. The sensitivity of the load ratio, P/p, on the ratio for vertical stresses can be observed by comparing the surfaces shown in Figures A-2 and A-3. Two ratios of load are considered: P/p = 50 and 100 [L], which correspond to one-half and one times the length of the x- and z-axes (horizontal and vertical distances, respectively) in Figures A-2 and A-3. The ratio of vertical (or shear) stresses extends from between 0 and 1 at great distances from the source and approaches infinity at locations near the

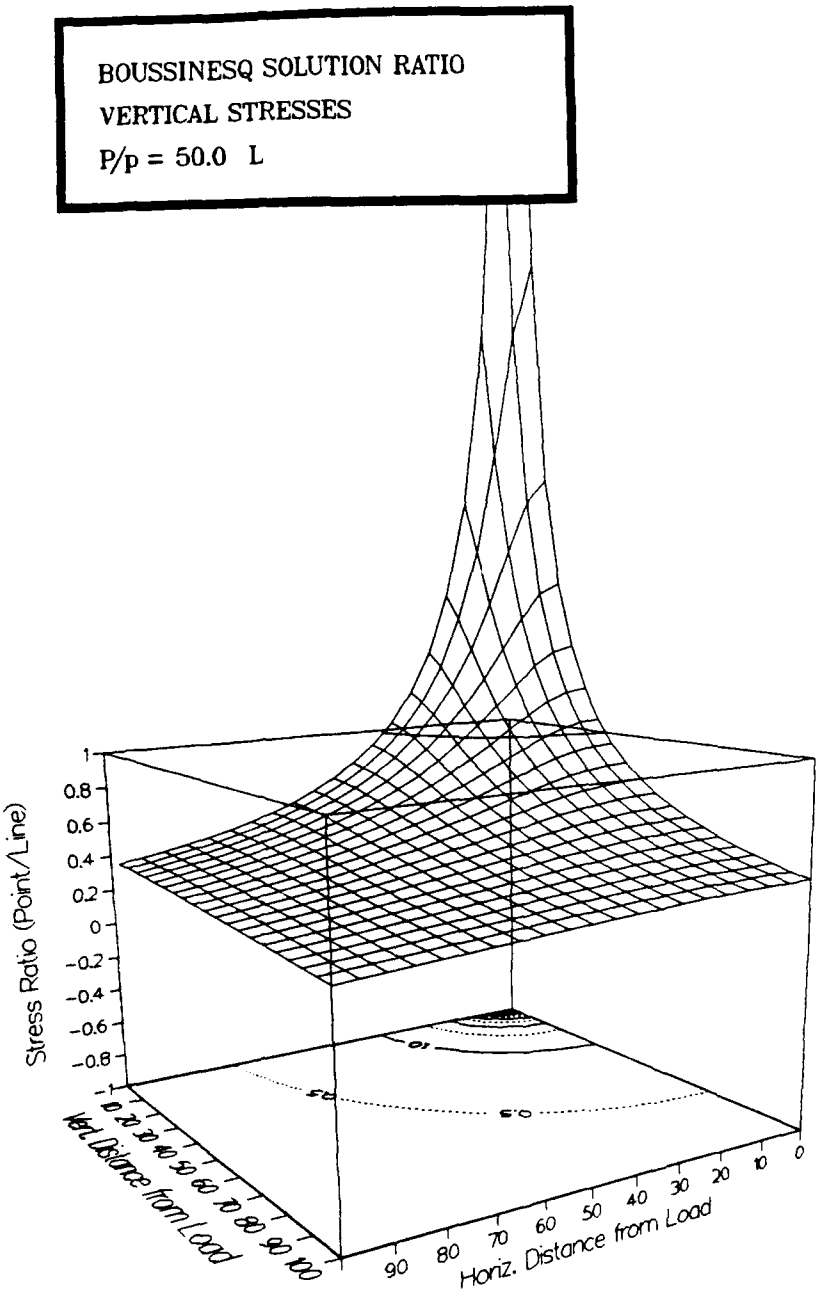


Figure A-2. Ratio of vertical stresses for vertical point and line loads for P/p = 50

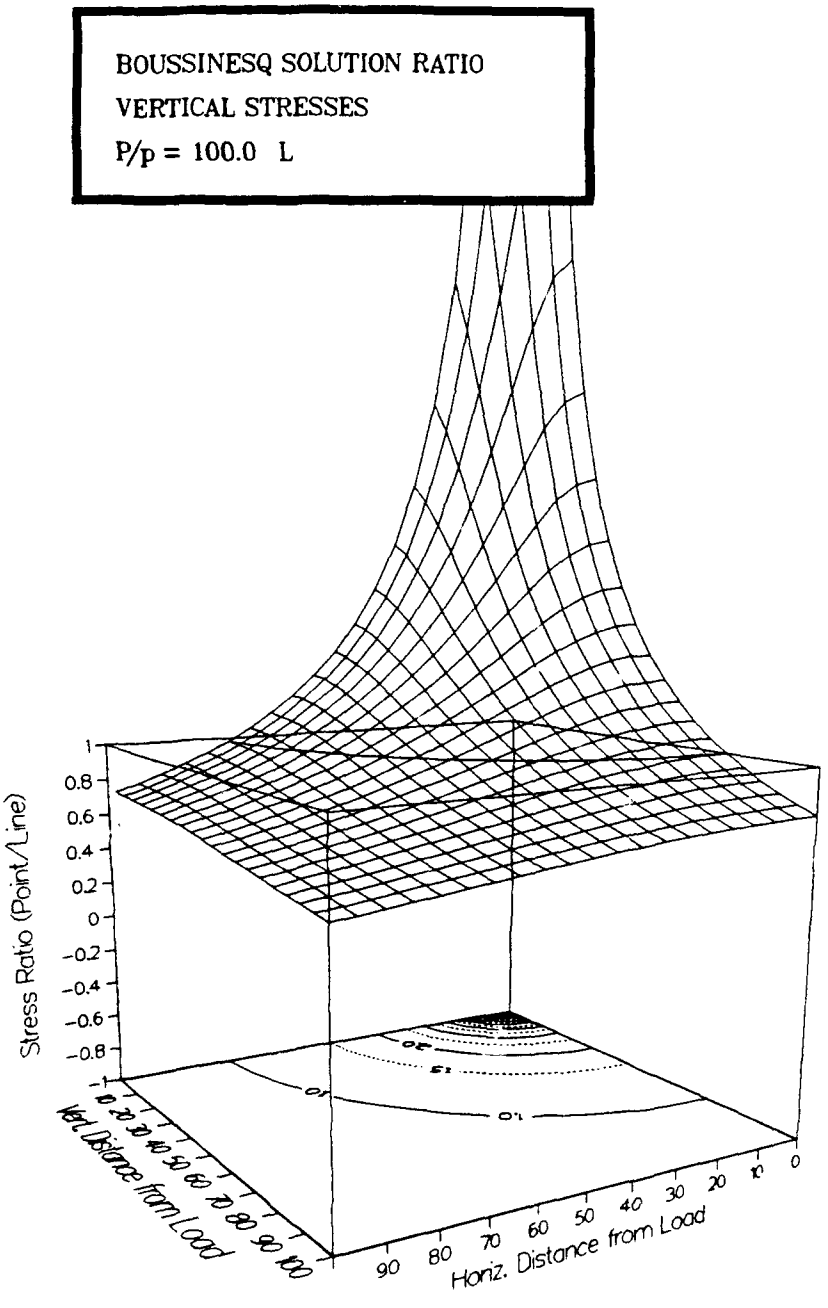


Figure A-3. Ratio of vertical stresses for vertical point and line loads for P/p = 100

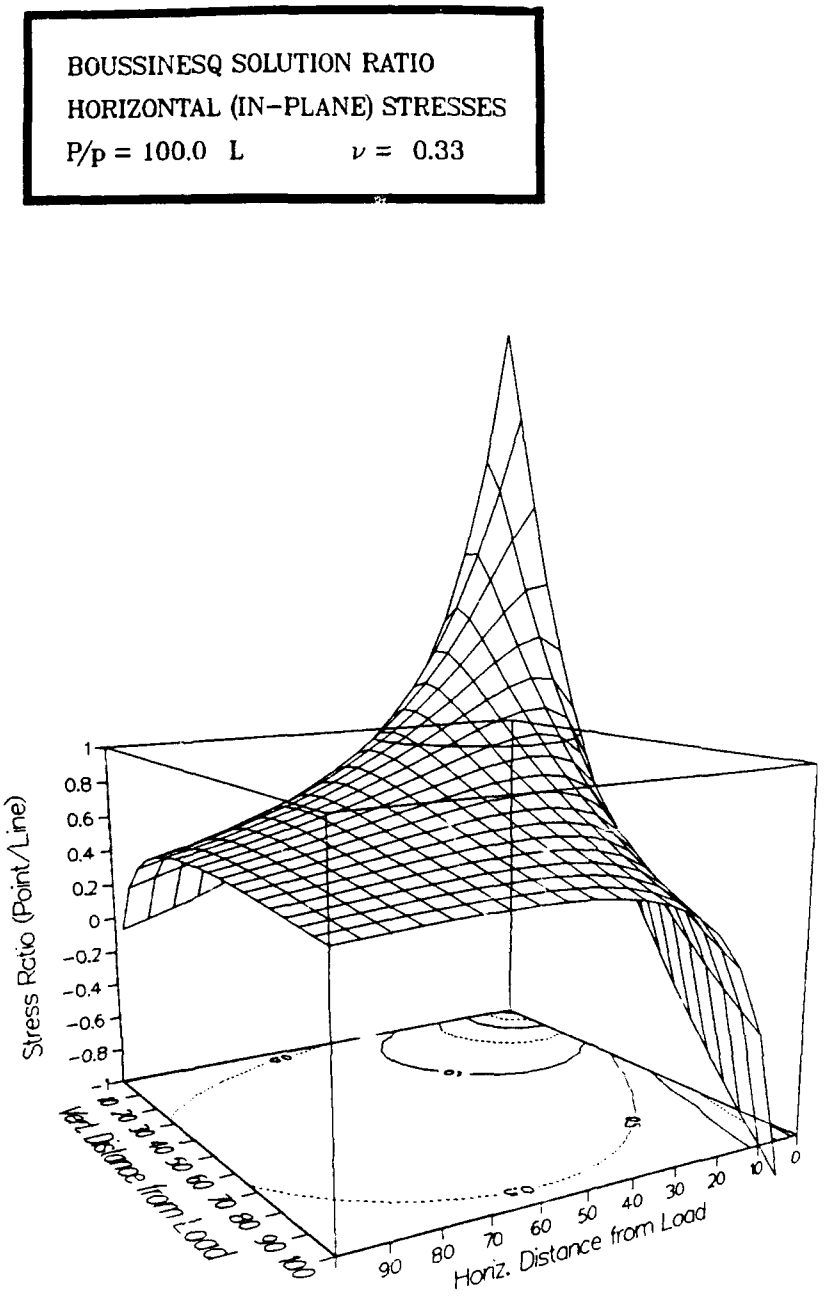


Figure A-4. Ratio of horizontal stresses for vertical point and line loads for P/p = 100 and ν = 0.33

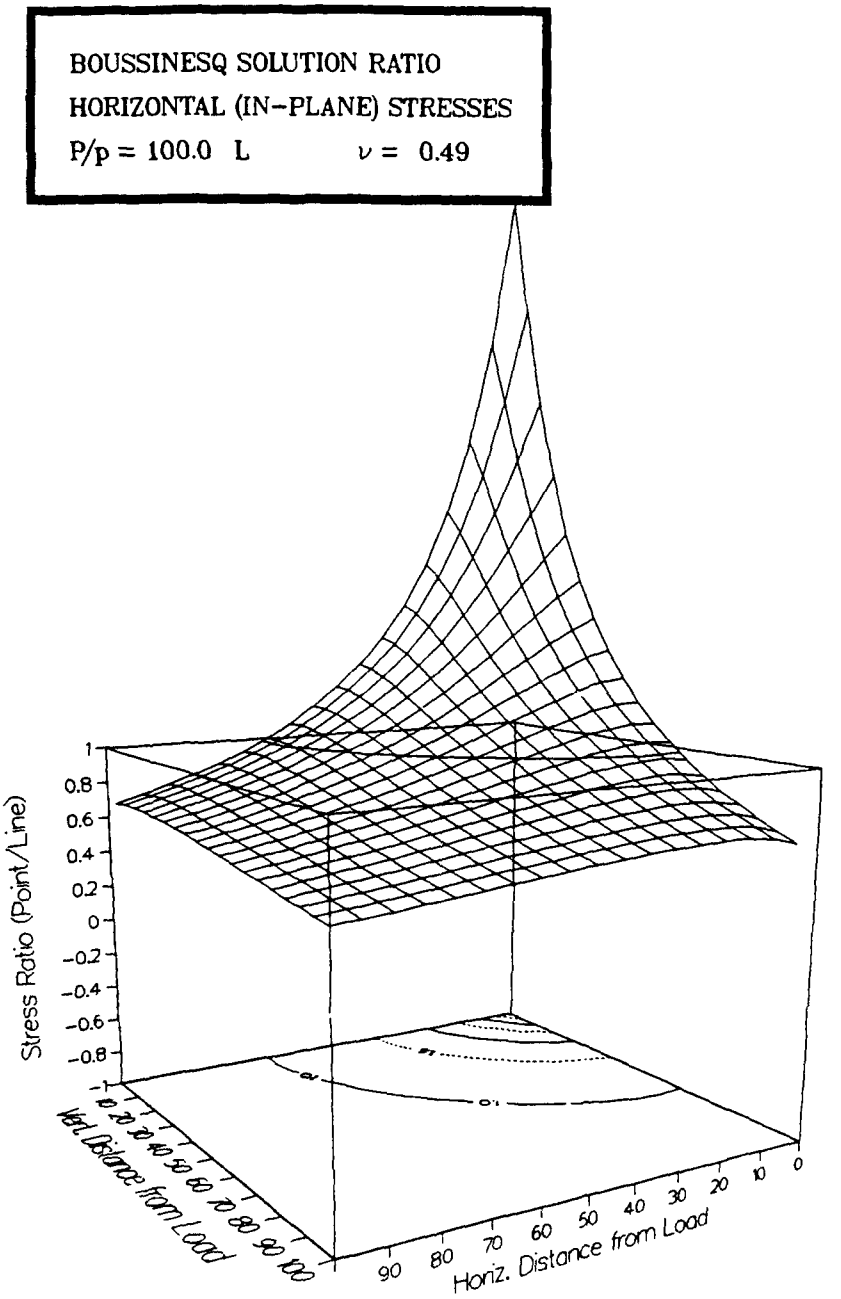


Figure A-5. Ratio of horizontal stresses for vertical point and line loads for P/p = 100 and ν = 0.49

source. For the load ratio of 50 [L] shown in Figure A-2, the ratio approaches infinity near the surface as the distance to the point of application approaches zero and approaches 0.38 at a distance of 100. So, within a range of distance of two orders of magnitude (1 to 100), the vertical stress for a point load varies two orders of magnitude (point load is 38 times to 0.38 times the vertical stress for a line load). The calculated stresses are equal at a radial distance on the vertical plane of 38. For a load ratio of 100 [L] shown in Figure A-3, the ratio ranges from 75 to 0.75 for distances of 1 to 100 with unity at a radial distance of 75.

A7. The gradient of the surfaces shown in Figures A-2 and A-3 is a function of the inverse of the square root of distance, $R^{-0.5}$. The gradient of the surface approaches infinity as $R \rightarrow 0$. Beyond the point where the ratio is unity, the gradient is low and approaches zero as $R \rightarrow P/p$. A lower or zero gradient is desirable for a stable matching of solutions. This suggests that if a load ratio were to be selected to best represent a point load using a two-dimensional solution for estimation purposes only, a large ratio of P/p be used while the radial distances of interest should be slightly less than, equal, or somewhat exceed the radial distance corresponding to unity.

A8. The sensitivity of ν on the ratio for horizontal stresses can be observed by comparing Figures A-4 and A-5. Two values of Poisson's ratio were used: 0.33 and 0.49. The surfaces representing horizontal stresses acting on a vertical plane indicate that the ratio extends to both positive and negative infinity. The ratio approaches positive infinity near the surface (as vertical distance approaches zero) near the point of application (as the horizontal distance approaches zero), negative infinity as the vertical or horizontal distance approaches zero, and a finite value between 0 and 1 for the remainder. The surface has lower gradients as ν increases.

A9. It should be clear that the stresses acting on a vertical 2-D plane produced by concentrated (point) loads are much different from those produced by a line load oriented perpendicular to the analysis plane for semi-infinite media. The results of numerical analyses that incorporate assumptions of plane strain to solve 3-D problems can be erroneous if solving for 3-D loads.

Horizontal loads

A10. Cerutti published explicit solutions to the stresses and displacements in a semi-infinite medium caused by a horizontal point load. The closed-form solutions for vertical stress, σ_v, horizontal stress, σ_h, and shear stress on a vertical plane, τ_{xz}, produced by a horizontal point load acting on the surface are:

$$\sigma_z^{pt} = \frac{3Qxz^2}{2\pi R^5} \tag{A10}$$

$$\sigma_x^{pt} = \frac{-Qx}{2\pi R^3}\left[\frac{-3x^2}{R^2} + \frac{(1-2\nu)R^2}{(R+z)^2}\right] \tag{A11}$$

$$\tau_{xz}^{pt} = \frac{3Qx^2z}{2\pi R^5} \tag{A12}$$

where

Q = magnitude of horizontal point load [F]
ν = Poisson's ratio

Notice that only Equation A11 is a function of a material property, ν.

A11. Equations A10 through A12 can be integrated to obtain explicit solutions caused by an infinite horizontal line load as reported by Poulos and Davis (1972). The variables of the problem are shown in Figure A-1. The closed-form solutions for a horizontal line load acting on the surface are:

$$\sigma_z^{ln} = \frac{2qx^2z}{\pi R^4} \tag{A13}$$

$$\sigma_x^{ln} = \frac{2qx^3}{\pi R^4} \tag{A14}$$

$$\tau_{xz}^{ln} = \frac{2qx^2z}{\pi R^4} \tag{A15}$$

where

q = magnitude of horizontal line load [F/L]

None of the three equations presented for horizontal line loads is a function of material properties.

A12. The explicit solutions for horizontal point and line loads were used to show the importance of correctly matching the type of load to the type of problem. The ratios of stresses acting on a vertical plane for point and lines loads are:

$$\frac{\sigma_z^{pt}}{\sigma_z^{ln}} = \frac{3}{4R}\frac{Q}{q} \tag{A16}$$

$$\frac{\sigma_x^{pt}}{\sigma_x^{ln}} = \frac{R^3}{4x^2}\left[\frac{3x^2}{R^4} - \frac{(1-2\nu)}{(R+z)^2}\right]\frac{Q}{q} \tag{A17}$$

$$\frac{\tau_{xz}^{pt}}{\tau_{xz}^{ln}} = \frac{3}{4R}\frac{Q}{q} \tag{A18}$$

The ratios for vertical stress and shear stress are again equivalent and are also equivalent to the stress ratios for vertical loads (Equations 7 and 9 and Figures A-2 and A-3). The relation produced from the ratio for horizontal stresses (Equation 17) caused by the horizontal load is only slightly different from that for vertical loads (Equation 8) and is plotted in Figure A-6 for $\nu = 0.33$.

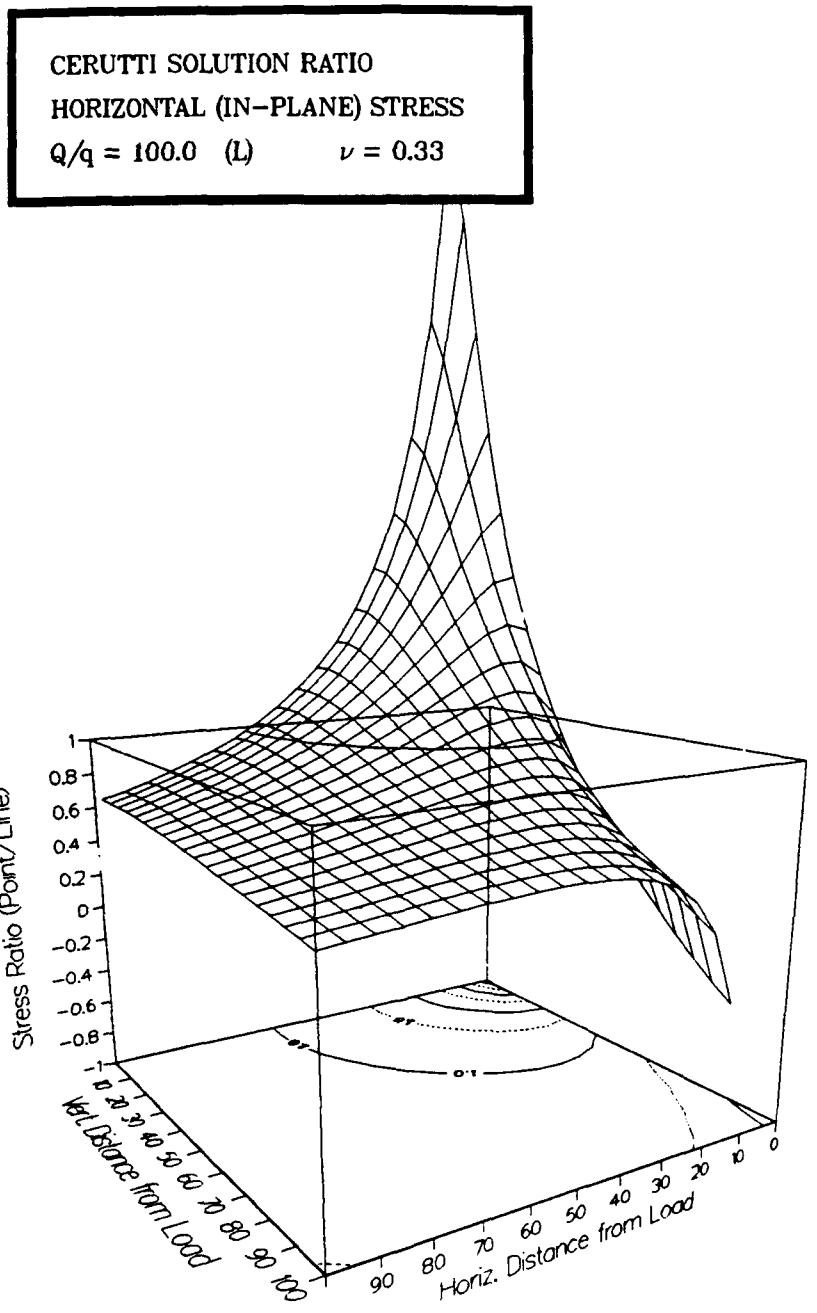


Figure A-6. Ratio of horizontal stresses for horizontal point and line loads for Q/q = 100 and ν = 0.33

APPENDIX B:
DERIVATION OF SYSTEM OF EQUATIONS
FOR THE FINITE ELEMENT METHOD

B1. The stress equilibrium equations are satisfied through the principle of virtual work (Zienkiewicz and Taylor 1989). Assuming that small virtual displacements, δU, occur at the nodes, virtual displacements within the element are obtained by:

$$\delta u = \mathbf{N}\, \delta U \tag{B1}$$

where

$\mathbf{N}$ = matrix of interpolation functions

Corresponding virtual strains are obtained by:

$$\delta \varepsilon = \mathbf{B}\, \delta U \tag{B2}$$

where

$\mathbf{B}$ = matrix of partial derivatives of shape functions

And the virtual strain energy, δU, is evaluated as:

$$\delta U = \int_v \delta \varepsilon^T \sigma \, dv \tag{B3}$$

where

ε^T is the transpose (vector) matrix of ε

"v" designates integration over the volume

The external virtual work, δW, done by the nodal forces, $\mathbf{P}$, and body forces, $\mathbf{b}$, can be defined as:

$$\delta W = \delta U^T \mathbf{P} + \int_v \delta u^T \mathbf{b} \, dv \tag{B4}$$

Applying the principle of virtual work (internal virtual work equals external virtual work):

$$\delta U = \delta W \tag{B5}$$

substituting:

$$\int_v \delta \varepsilon_T \, \sigma \, dv = \delta U^T \mathbf{P} + \int_v \delta u^T \mathbf{b} \, dv \tag{B6}$$

Through substitution and the use of transpose vectors, the following is obtained:

$$\delta U^T \left(\int_v B^T D B \, dv \right) U = \delta U^T P + \delta U^T \int_v N^T b \, dv \tag{B7}$$

and

$$\left(\int_v B^T D B \, dv \right) U = P + \int_v N^T b \, dv \tag{B8}$$

B2. Assuming that the only contribution of the body force is from the inertial effects and using d'Alembert's principle:

$$b = -\rho \ddot{u} \tag{B9}$$

where

ρ = mass density

$\ddot{u}$ = are accelerations for Cartesian components

Substituting equation B1 into B9:

$$b = -\rho N \ddot{U} \tag{B10}$$

Further substitution leads to:

$$\left(\int_v B^T D B \, dv \right) U = P - \left(\int_v \rho N^T N \, dv \right) \ddot{U} \tag{B11}$$

APPENDIX C:

FINITE ELEMENT PROGRAM *vib3*

C1. The following text is a listing of the computer program *vib3* written in FORTRAN.

```
      PROGRAM VIB3_4
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Program VIB3 is a finite element code written to solve the     C
C     the problem of wave propagation produced by 3-D loads acting   C
C     in 2-D soil/geologic systems.  The stiffness matrix for        C
C     each 3-D element is condensed into an equivalent 2-D           C
C     (dynamic) stiffness.                                           C
C                                                                     C
C     The methodology f'r solving the problem was provided by        C
C     Prof. Jose Roes-  and Dr. Vincent Kang at the University       C
C     of TEXAS at Aust.n as part of their research to solve a        C
C     similar problem for pavement systems (completed May 1990).     C
C     The basic structure and frontal solver used                    C
C     were initially provided by Professor John Tassoulas at         C
C     the University of TEXAS at Austin for a graduate course on     C
C     the finite element method.                                     C
C                                                                     C
C     The program has been specialized for a 16-node isoparametric   C
C     element that is quadratic in the x and z directions and linear C
C     in the y direction (direction of condensation).                C
C                                                                     C
C     At the present, this code is written for single frequency      C
C     harmonic excitations.  Transient and multi-frequency exci-     C
C     tations will require a FFT from the time domain to th-         C
C     temporal frequency domain.                                     C
C                                                                     C
C     VIB3_4.F designates memory requirements consistent with        C
C     batch queue "prime 4_."                                        C
C                                                                     C
C     VIB3_4.F uses two single-subscripted arrays to store all       C
C     massive information (A&C). Indicies point to the beginning of  C
C     various arrays.                                                C
C                                                                     C
C     This program was last updated on 24 August 1992.               C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C  VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     The limitations on the program input are:                      C
C                                                                     C
C         Use with 3-D, 16-node element only                         C
C         No (static) surface loads (see MODIF)                      C
C         Zero body forces (see MODIF)                               C
C         Maximum of 50 (select) nodes (Dim. in MAIN & DATAIN)       C
C         Maximum of 3000 nodes for multi-D plotting (4 Mword limit) C
C         Maximum of 10 frequencies of interest                      C
C         Maximum of 512=NM (Dimension and IND(17))                  C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C  VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     The input parameter list (in general) is:                      C
C                                                                     C
C       Location          Variables                                  C
C       --------          ---------                                  C
C       MAIN              TITLE1 (Up to 50 characters)               C
C       MAIN              TITLE2 (Up to 50 characters)               C
C       MAIN              TITLE3 (Up to 50 characters)               C
```

```
C       MAIN                INCHECK, INOUT, IPRINT, INPLOT              C
C                             where: INCHECK=0  check input only        C
C                                           1   full execution          C
C                                    INOUT=0    minimal output          C
C                                          1    full output             C
C                                    IPRINT=0   no solver information   C
C                                           1   minimal solver info.    C
C                                         =>2   full solver info.       C
C                                    IPLOT=0    no dynamic plot files   C
C                                          1    dynamic plot files      C
C       MAIN                NN2,NUMEL,NOMAT                             C
C       MAIN                MA                                          C
C       MAIN                NF,DY,NM,NSLCT                              C
C       MAIN          (loop) FR (Hz)                                    C
C       MAIN                TOLSM(SM)                                   C
C       DATAIN        (loop) NODE#, X,Y,Z                               C
C       DATAIN        (loop) NS,IDIR,IVAR                               C
C       DATAIN               NELX,NELZ                                  C
C       DATAIN        (loop) EL#,MAT,ICONN                              C
C       ELP3D         (loop) MAT,G,PR,DAMP,RO                           C
C                             where:  DAMP  (decimal)                   C
C       YLOAD                PMAG,YLDIS                                 C
C                             where:  PMAG*YLDIS=0.5 for normalized     C
C                                     load (total load = 1)             C
C       MAIN                 XL,XR                                      C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C  VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F  C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C   The units for this program are universal.  Consistent units        C
C   are as follows:                                                     C
C                                                                       C
C                          METRIC             SI                      C
C                          ------            ------                   C
C   Shear modulus            Pa                psf                    C
C   Mass density           kg/m**3          (lb-s**2)/ft               C
C                                            = slugs/ft**3              C
C                                            = pcf/32.17                C
C   Displacement             m                 ft                     C
C                                                                       C
C         Damping ratio is in decimal form.                             C
C         Frequency must be in Hz.                                      C
C                                                                       C
C   The following conversion factors are provided for convenience:      C
C                                                                       C
C         multiply              by              to   get                C
C         --------           ----------        -------------            C
C           psf                 47.88026           Pa (N/m**2)          C
C           bar                  0.00001           Pa                   C
C        slugs/ft**3           515.3              kg/m**3              C
C            ft                  0.3048              m                  C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C  VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F  C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C     Tape (unformatted) Library:                                       C
C                                                                       C
C     No.         Contents                                            C
C    *****    ***************************                             C
C       7       (TAPEB)                                               C
C       8       Displacement vector (TAPEU)                           C
C       9       Load vector (TAPEL)                                   C
C      10       LHS & RHS for solution                                C
C      12       Original load vector                                  C
```

```
C     Tape (formatted) library:                                          C
C                                                                        C
C     15        Real components of select stiffness matrices             C
C     16        Imaginary components of select stiffness matrices        C
C     20        Primary output file (NTAPEO)                             C
C     22        Non-zero components of load vector(s)                    C
C     23        Input file for dynamic 2-D plotting (dview)              C
C     25        Input file for static plotting (y_plot & pltwave)        C
C                                                                        C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                        C
C  VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F   ...   VIB3.F  C
C                                                                        C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                        C
C     PRIMARY VARIABLE LIST:                                             C
C       (Variables used for solver listed under Subroutine SOLVE.)       C
C                                                                        C
C       A       Real/Integer (1-D) array used for most data.             C
C       AI      i                                                        C
C       C       Complex/Integer (1-D) array used for SOLVER              C
C       CFOR    Wavenumber-domain description of load in y-direction     C
C       CRHS    Complex RHS from dynamic, surface, & body forces         C
C       CRHSC   Condensed, complex RHS                                   C
C       CSM     Complex stiffness matrix (3-D)                           C
C       CSMC    Condensed, complex stiffness matrix (2-D)                C
C       CW      Complex amplitude of select nodes in z-direction         C
C       CWIM    Complex amplitude of select nodes for specific im        C
C       DAMP    Damping ratio (-)                                        C
C       DAMPC   Complex damping ratio                                    C
C       DM      Increment of wavenumber m (delta m)                      C
C       DY      Increment of spatial variable y (delta y)                C
C       FACT    Factor for inverse FFT                                   C
C       FR      Frequencies of interest                                  C
C       FOR     Spatial-domain description of load in y-direction        C
C       F2      Conversion factor (radians to degrees)                   C
C       G       Shear modulus (real)                                     C
C       GC      Complex shear modulus                                    C
C       ICONN   Element connectivity for 3-D mesh                        C
C       MA      Memory allocation for solver                             C
C       MAT     Material type index                                      C
C       NCH     NM+2                                                     C
C       NDOFE2  No. of degrees of freedom per elem., cond. mesh          C
C       NDOFN2  No. of degrees of freedom per node, cond. mesh           C
c       NDOFT2  No. of degrees of freedom (load vector), cond. mesh      C
C       NELX    No. of elements in x-direction, 3-D mesh                 C
C       NELZ    No. of elements in z-direction, 3-D mesh                 C
C       NF      No. of frequencies of interest                           C
C       NSLCT   No. of select nodes                                      C
C       NM      No. of m's for FFT                                       C
C       NM2     Half the NM's                                            C
C       NM2P1   NM2+1                                                    C
C       NM2P2   NM2+2                                                    C
C       NN      Number of nodes, 3-D mesh (1-element thick)              C
C       NNE2    Number of nodes per element, condensed mesh              C
C       NNT2    Number of nodes (total), condensed mesh                  C
C       NOMAT   Number of material types                                 C
C       NS      Nodes of interests                                       C
C       NUMEL   Number of elements, 3-D mesh                            C
C       NUMEL2  Number of elements, condensed mesh (=NUMEL)              C
C       OM      Operating frequency (rads/sec)                           C
C       P       Surface loads (zero)                                     C
C       PMAG    Magnitude of dynamic load                                C
C       PR      Poisson's ratio                                          C
C       RHS     Body forces (real)                                       C
C       RO      Mass density                                             C
C       TM      Value of m (real) in DO loop                               C
C       TOLSM   Tolerance for symmetry of stiffness matrices             C
C       X       Coordinates of nodes (3-D)                               C
```

```
C       XL      X-coordinate at left edge of load                        C
C       XR      X-coordinate at right edge of load                       C
C       XMAX    Max. extent of 3-D mesh in x-direction (-XMAX to XMAX)C
C       YLDIS   Extent of load in y-direction from y=0                   C
C       YTOT    Distance for Fourier expansion (DY*NM)                   C
C       ZMAX    Max. extent of 3-D mesh in z-direction (0 to ZMAX)       C
C                                                                        C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION IA(70000),A(70000),IC(200000),FR(10),NYL(8),IVAR(50)
      COMPLEX C(200000),CW(514,3000),CWIM(514)
C
C     Dimension changes to arrays C and IC must also be made to
C     DATA NC, and arrays A and IA in SUBROUTINE SOLOUT.
C
      COMPLEX E1,E2,AI,AI1,CPD
      EQUIVALENCE (A(1),IA(1)),(C(1),IC(1))
      CHARACTER*50 TITLE1,TITLE2,TITLE3
      COMMON /BIG/ BIG
      COMMON/CNTL/ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,MA,
     .            IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .            IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .            MELEM,MFWR,MB,MDOF,MFW,MLDEST
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /FLAGS/ IND(22)
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /INTERPF/ AN1D(16,18),DNDXI(3,16,18),W(18)
      COMMON /MAX/ XMAX,ZMAX
      COMMON /MESH/ NELX,NELZ
      COMMON /REALA/ A
      COMMON /TOL/ TOLSM
      COMMON /UNIT/ NTAPE,NTAPEO
      COMMON /WORK/ C
      DATA NA/70000/
      DATA NC/200000/
      DATA NYL/4,8,16,32,64,128,256,512/
C>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C                                                                        !
C     Constants for FRONTAL SOLVER:                                      !
C        ISYM=1:  Symmetric stiffness matrix                             !
C            =2:  Unsymmetric resolution inactivated                     !
C            =3:  Unsymmetric resolution                                 !
C        IRESOL=0:  CALL COMPL (lower triangle of stiffness matrix)      !
C               =1:  CALL RESOL (full stiffness matrix)                  !
C        NRHS=0:  Resolution with zero RHS                               !
C            =1:  One RHS                                                !
C            =i:  i RHS's                                                !
C                                                                        !
C     Solver output index-                                               !
C         IPRINT=0: no solver output                                     !
C         IPRINT=1: general output                                       !
C         IPRINT>=2: full information (pivot infor.)                     !
C                                                                        !
          ISYM=1                                                         !
          IRESOL=0                                                       !
          NRHS=1                                                         !
C         IPRINT=1                                                       !
C                                                                        !
          NTAPEB=7                                                       !
          NTAPEU=8                                                       !
C                                                                        !
C         NTAPEL is used only for NRHS > 1                               !
C                                                                        !
          NTAPEL=9                                                       !
          NTAPE=10                                                       !
          NTAPEO=20                                                      !
C>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
      OPEN(15,FILE='STIFF_R',STATUS='UNKNOWN')
      OPEN(16,FILE='STIFF_I',STATUS='UNKNOWN')
      OPEN(NTAPEO,FILE='OUTPUT',STATUS='UNKNOWN')
      OPEN(22,FILE='LOADFFT',STATUS='UNKNOWN')
      OPEN(23,FILE='dplot_2d.in',STATUS='UNKNOWN')
      OPEN(25,FILE='plot.in',STATUS='UNKNOWN')
C
C     Basic information about the analysis
C
      NDIM=3
      MNNE=16
      MNDOFN=3
      MDOFE=48
      BIG=1.0E50
      WRITE(NTAPEO,5)
    5 FORMAT('******************** PROGRAM VIB3 ********************',//,
     * 1x,'  This program was written to solve for dynamic displace-',/,
     * 1x,'ments in complex soil/geologic media using a 2-D finite  ',/,
     * 1x,'element formulation.  The formulation assumes planar     ',/,
     * 1x,'geometry and material properties in the out-of-plane   ',/,
     * 1x,'direction and a harmonic source acting on the surface.',//,
     * 1x,'  This program was written by David Sykora, at US Army',/,
     * 1x,'Engineer Waterways Experiment Station (WES), Vicksburg,',/,
     * 1x,'MS, under sponsorship of ILIR program (FY90-92).',//,
     * 1x,'  Prof. Jose Roesset, Univ. of Texas at Austin, developed',/,
     * 1x,'the condensation procedure used in the formulation as ',/,
     * 1x,'successfully implemented by Dr. Kang (1990) for pave- ',/,
     * 1x,'ment systems.  Solver subroutines, the FFT routines, and ',/,
     * 1x,'the basic framework of the finite element program were  ',/,
     * 1x,'obtained from Profs. Roesset and Tassoulas, UT.  ',//,
     * 1x,'THIS SOFTWARE IS DISTRIBUTED AS IS AND WITHOUT WARRANTY',/,
     * 1x,'AS TO PERFORMANCE.  THE USER MUST ASSUME THE RISK OF',/,
     * 1x,'USING THIS SOFTWARE!',//)
      READ(5,6) TITLE1
      READ(5,6) TITLE2
      READ(5,6) TITLE3
      READ(5,*) INCHECK,INOUT,IPRINT,INPLOT
      READ(5,*) NNT2,NUMEL,NOMAT
      NN=NNT2*2
      READ(5,*) MA
      READ(5,*) NF,DY,NM,NSLCT
      DO 2 I=1,8
        IF(NM.EQ.NYL(I)) GOTO 3
        IF(I.EQ.8) THEN
          WRITE(6,*)'INCORRECT INPUT VALUE OF NM'
          STOP
        ENDIF
    2 CONTINUE
    3 CONTINUE
      WRITE(NTAPEO,6)TITLE1
      WRITE(NTAPEO,6)TITLE2
      WRITE(NTAPEO,6)TITLE3
      WRITE(NTAPEO,15) NM,DY,NOMAT,NSLCT,BIG,MA
      WRITE(NTAPEO,16) NDIM,NUMEL,NN,MNNE,MNDOFN,MDOFE
    6 FORMAT(A50)
   15 FORMAT(//,'*****PROBLEM PARAMETERS:',//,
     1       1X,'NUMBER OF TERMS FOR FFT:      ',I6,/,
     2       1X,'INCREMENT OF Y (DY):          ',F6.2,/,
     1       1X,'NUMBER OF MATERIAL TYPES:     ',I6,/,
     2       1X,'NUMBER OF NODES OF INTEREST: ',I6,/,
     3       1X,'"BIG":                    ',E12.7,/,
     4       1X,'MAXIMUM ARRAY ALLOCATION:   ',I7,//)
   16 FORMAT(/,'    3-D MESH:',//,
     1       1X,'NUMBER OF DIMENSIONS:         ',I6,/,
     2       1X,'NUMBER OF ELEMENTS:           ',I6,/,
     3       1X,'NUMBER OF NODES:              ',I6,/,
     4       1X,'NUMBER OF NODES/ELEMENT:      ',I6,/,
     5       1X,'DEGREES OF FREEDOM/NODE: ',I10,/,
     6       1X,'DEGREES OF FREEDOM/ELEMENT:  ',I6,//)
```

```
C     Parameters for condensed elements
C
      NUMEL2=NUMEL
      NNE2=MNNE/2
      NDOFN2=MNDOFN
      NDOFE2=NNE2*NDOFN2
      NDOFT2=NUMEL2*NDOFE2
      WRITE(NTAPEO,17) NDIM,NUMEL2,NNT2,NNE2,NDOFN2,NDOFE2
      IF(INPLOT.NE.0) THEN
        WRITE(23,*)NNT2,NUMEL2
      ELSEIF(INPLOT.EQ.0) THEN
        WRITE(23,*)'No 2-D dynamic plot file created (INPLOT=0).'
      ENDIF
   17 FORMAT(/,'    CONDENSED MESH:',//,
     1       1X,'NUMBER OF DIMENSIONS:        ',I6,/,
     2       1X,'NUMBER OF ELEMENTS:          ',I6,/,
     3       1X,'NUMBER OF NODES:             ',I6,/,
     4       1X,'NUMBER OF NODES/ELEMENT:     ',I6,/,
     5       1X,'DEGREES OF FREEDOM/NODE: ',I10,/,
     6       1X,'DEGREES OF FREEDOM/ELEMENT:  ',I6,//)
C >>>>>>>>>>>>>>>>>>>> STORAGE ALLOCATION >>>>>>>>>>>>>>>>>>>>>>>>>>>>
C                                                                      !
C********* BELOW CORRESPOND TO ARRAY A (real/integer) ****************!
C                                                                      !
      IND(1)=1                                                         !
C.....X (3-D nodal coordinates)                                        !
      IND(2)=IND(1)+3*NN                                               !
C.....G (real shear modulus)                                           !
      IND(3)=IND(2)+NOMAT                                              !
C.....IS (boundary conditions)                                         !
      IND(4)=IND(3)+3*NN                                               !
C.....FR (frequencies of interest)                                     !
      IND(5)=IND(4)+NF                                                 !
C.....NS (nodes of interest)                                           !
      IND(6)=IND(5)+NSLCT+7                                            !
C.....IDIR                                                             !
      IND(7)=IND(6)+NSLCT                                              !
C.....ICONN (element connectivities)                                   !
      IND(8)=IND(7)+NUMEL*MNNE                                         !
c.....MAT (material type index)                                        !
      IND(9)=IND(8)+NUMEL                                              !
C.....RO (mass density)                                                !
      IND(10)=IND(9)+NOMAT                                             !
C                                                                      !
C****** BELOW CORRESPOND TO ARRAY C (complex/integer) ****************!
      IND(11)=1                                                        !
C.....IPREP (used in PREFNT)                                           !
      IND(12)=IND(11)+2*(NUMEL2*NNE2+NNE2)                             !
C.....IDEST (used in DESVEC)                                           !
      IND(13)=IND(12)+NUMEL2*NNE2                                      !
C.....COMPLEX ARRAY FOR SUBROUTINE SOLVE                               !
      IND(14)=IND(13)+MA                                               !
C.....COMPLEX ARRAY FOR 2-D SOLUTION (amplitude=f(x,m,z,OM))           !
      IND(15)=IND(14)+NDOFN2*NNT2                                      !
C.....Complex load vector components                                   !
      IND(16)=IND(15)+NDOFT2                                           !
C.....Material constants                                               !
      IND(17)=IND(16)+3*NOMAT                                          !
C.....CFOR                                                             !
      IND(18)=IND(17)+513                                              !
C.....                                                                 !
      IND(19)=IND(18)                                                  !
C.....                                                                 !
      IND(20)=IND(19)                                                  !
C.....                                                                 !
      IND(21)=IND(20)                                                  !
C.....                                                                 !
      IND(22)=IND(21)                                                  !
C >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
      WRITE(6,29)
   29 FORMAT(5(/),25X, 'V     V  I   BBBB    33333',/,
     +           25X, 'V     V  I   B   B       3',/,
     +           25X, ' V   V   I   B   B     3 ',/,
     +           25X, ' V   V   I   BBBB    333 ',/,
     +           25X, '  V V    I   B   B       3',/,
     +           25X, '  V V    I   B   B       3',/,
     +           25X, '   V     I   BBBB    333 ',////)
C
C     Check maximum storage (for Array A)
C
      MAXA=IND(10)-1
      IF(MAXA.GT.NA) THEN
        WRITE(6,30) MAXA
        IF(INOUT.EQ.1) WRITE(NTAPEO,30) MAXA
   30   FORMAT(1X,'INSUFFICIENT MEMORY LOCATIONS',/,
     *         1X,'REQUIRED LENGTH OF ARRAY A:',1X,I7,/)
        STOP
      ELSE
        IF(INOUT.EQ.1) WRITE(NTAPEO,35) NA-MAXA
        WRITE(6,35) NA-MAXA
   35   FORMAT(1X,
     +     'NUMBER OF UNUSED MEMORY WORDS IN ARRAY A:',1X,I7,/)
      ENDIF
C
C     Check maximum storage (for Array C)
C
      MAXC=IND(22)-1
      IF(MAXC.GT.NC) THEN
        WRITE(6,31) MAXC
        IF(INOUT.EQ.1)WRITE(NTAPEO,31) MAXC
   31   FORMAT(1X,'INSUFFICIENT MEMORY LOCATIONS',/,
     *         1X,'REQUIRED LENGTH OF ARRAY C:',1X,I7,/)
        STOP
      ELSE
        IF(INOUT.EQ.1)WRITE(NTAPEO,36) NC-MAXC
        WRITE(6,36) NC-MAXC
   36   FORMAT(1X,
     +     'NUMBER OF UNUSED MEMORY WORDS IN ARRAY C:',1X,I7,/)
      ENDIF
C >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
      WRITE(NTAPEO,49)
   49 FORMAT(/,1X,'FREQUENCIES OF INTEREST (Hz): ',/)
      DO 50 IF=1,NF
   50 READ(5,*) FR(IF)
      DO 51 IF=1,NF,6
   51 WRITE(NTAPEO,55) (FR(I),I=IF,IF+5)
   55 FORMAT(1X,6F10.3)
      READ(5,*) TOLSM
      IF(INOUT.EQ.1)WRITE(NTAPEO,59)
      IF(INOUT.EQ.1)WRITE(NTAPEO,60) TOLSM
   59 FORMAT(//,1X,'SPECIFIED TOLERANCES- ',/)
   60 FORMAT(1X,'STIFFNESS MATRIX SYMMETRY: ',E15.4,3X,'[F/L]',//)
      CALL DATAIN(NUMEL,A(IND(1)),IA(IND(7)),IA(IND(3)),IA(IND(8)),
     2                  IA(IND(5)),IA(IND(6)),IC(IND(11)),DY,IVAR)
      CALL ELP3D(C(IND(16)),A(IND(9)),A(IND(2)))
      NMP2=NM+2
      NM2=NM/2
      NM2P1=NM2+1
      NM2P2=NM2P1+1
      NCH=2*NM2P1
      YTOT=NM*DY
      PI=4.0*ATAN(1.0)
      DM=2.*PI/YTOT
      IF(INPLOT.EQ.1) THEN
        WRITE(23,*)NM2P1,DY
      ENDIF
      WRITE(25,*)NM2P1,DY,NSLCT
      AI=(0.,1.)
```

```
      CALL YLOAD(C(IND(17)),DY,NM)
      CALL FOUR2(C(IND(17)),NM,1,-1,1)
C   ---------------------------------------------------------------
      READ(5,*)XL,XR
      WRITE(NTAPE0,80) XL,XR
   80 FORMAT(//,1X,'Left-most extent of load in x-direction = ',F7.2,/,
     2        1X,'Right-most extent of load in x-direction = ',F6.2,//)
C
C     Begin Do loop for each frequency (rads/sec) of interest
C
      CALL DNISO16YF
      DO 100 IF=1,NF
        DO 101 I=1,514
          DO 101 J=1,3000
  101   CW(I,J)=(0.,0.)
        OM=2.*PI*FR(IF)
        WRITE(6,105) OM,FR(IF)
        WRITE(22,105) OM,FR(IF)
        IF(INPLOT.EQ.1) THEN
          WRITE(23,*) OM
        ENDIF
        WRITE(25,*) FR(IF)
  105   FORMAT(1X,'FREQUENCY= ',F6.2,' rads/sec = ',F6.2,' Hz',/)
C
C       Begin DO loop for each wavenumber, m (y-direction)
C       Since the loading and geometry are symmetric about y=0,
C       the displacements in the space domain should be symmetric
C       about y=0.  Therefore, only non-negative wavenumbers need
C       be considered and negative wavenumbers will be duplicated
C       later.
C
        CALL XLOAD(A(IND(1)),IA(IND(7)),XL,XR,NUMEL,C(IND(15)))
        IF(INPLOT.EQ.1) THEN
          WRITE(23,*)XL,XR
        ENDIF
        WRITE(25,*)XL,XR
        DO 200 IM=1,NM2P1
C
C         Read load information in x-direction & calculate nodal forces
C
          TM=(IM-1)*DM
          AI1=AI*TM*DY
          E1=CEXP(-1.*AI1)
          E2=CEXP(AI1)
C
C         Create load vector
C
          REWIND 10
          REWIND 12
          DO 210 I=1,NDOFT2
            WRITE(12) C(IND(15)+I-1)*C(IND(17)+IM-1)
  210     CONTINUE
          CALL STIFF(A(IND(1)),IA(IND(3)),IA(IND(7)),
     2               IA(IND(8)),C(IND(16)),A(IND(9)),A(IND(2)),
     3               NUMEL,OM,E1,E2,IM)
          IF(INCHECK.EQ.0) STOP
          REWIND 10
          REWIND 12
          CALL PREFNT(NNE2,IC(IND(11)),MS,MU,MR)
C
C         Solve for maximum amplitudes at each node =f(x,m,z,om)
C
          CALL SOLVE(C(IND(13)),IM)
C
C         Store amplitudes at select nodes for inverse FFT
C         (Always store NSLCT plus necessary values for dview)
C
```

```
          K=0
          DO 500 J=1,NSLCT
            K=K+1
            CW(IM,K)=C(IND(14)+3*(IA(IND(5)+J-1)-1)+IA(IND(6)+J-1)-1)
  500     CONTINUE
          IF(INPLOT.EQ.1) THEN
            DO 501 J=1,2*NELX+1
              K=K+1
              CW(IM,K)=C(IND(14)+3*(J-1))                              ! X
              K=K+1
              CW(IM,K)=C(IND(14)+3*(J-1)+1)                            ! Y
  501       CONTINUE
          ELSEIF(INPLOT.EQ.2) THEN
            DO 502 J=1,NNT2
              K=K+1
              CW(IM,K)=C(IND(14)+3*(J-1))                              ! X
              K=K+1
              CW(IM,K)=C(IND(14)+3*(J-1)+2)                            ! Z
  502       CONTINUE
          ELSEIF(INPLOT.EQ.3) THEN
            DO 503 J=1,2*NELZ+1
              K=K+1
              CW(IM,K)=C(IND(14)+3*(1+MOD((J-1),2)*(2*NELX+1)
     2                 +(J/2)*(3*NELX+2))+1)                           ! Y
              K=K+1
              CW(IM,K)=C(IND(14)+3*(1+MOD((J-1),2)*(2*NELX+1)
     2                 +(J/2)*(3*NELX+2))+2)                           ! Z
  503       CONTINUE
          ENDIF
          NIFTS=K
  200   CONTINUE
C
C       Calculate complex amplitudes at each node = f(x,y,z,om)
C
        WRITE(NTAPEO,401)FR(IF)
 401    FORMAT(///,'*****OUTPUT (DISPLACEMENTS)',//,
     1   1X,'CALCULATED AMPLITUDES AT: ',F8.2,' Hz')
        FACT=1./FLOAT(NM)
        F2=180./PI
C
C       Begin loop to IFT displacements for each node of interest
C
        DO 600 I=1,NSLCT
          DO 601 II=1,514
  601     CWIM(II)=(0.,0.)
C
C         Duplicate amplitudes for m's from 0 to NM2P1 (negative m's).
C
          DO 620 IM=NM2P2,NM
            IIM=NCH-IM
  620     CW(IM,I)=CW(IIM,I)
C
          DO 625 IM=1,NM
  625     CWIM(IM)=CW(IM,I)
C
C         Inverse FFT leaving amplitudes complex.
C
          CALL FOUR2(CWIM,NM,1,+1,1)
          WRITE(NTAPEO,635) IA(IND(5)+I-1),
     +        REAL(A(IND(1)+(IA(IND(5)+I-1))*3-3)),
     +        REAL(A(IND(1)+(IA(IND(5)+I-1))*3-1))
          IF(IA(IND(6)+I-1).EQ.1) WRITE(NTAPEO,627)
          IF(IA(IND(6)+I-1).EQ.2) WRITE(NTAPEO,628)
          IF(IA(IND(6)+I-1).EQ.3) WRITE(NTAPEO,629)
          WRITE(NTAPEO,636)
          WRITE(25,*) IA(IND(5)+I-1),
     +      REAL(A(IND(1)+(IA(IND(5)+I-1))*3-3)),
     +      REAL(A(IND(1)+(IA(IND(5)+I-1))*3-1)),
     +      IA(IND(6)+I-1),IVAR(I)
```

```
          DO 640 IM=1,NM2P1
            Y=(IM-1)*DY
            CWIM(IM)=FACT*CWIM(IM)
            AMPL=SQRT(REAL(CWIM(IM))**2+AIMAG(CWIM(IM))**2)
            PHAS=ATAN2(AIMAG(CWIM(IM)),REAL(CWIM(IM)))*F2
            WRITE(25,645) Y,CWIM(IM),AMPL,PHAS
            WRITE(NTAPEO,645) Y,CWIM(IM),AMPL,PHAS
            IF(IVAR(I).EQ.0) GOTO 641
 640      CONTINUE
 641      CONTINUE
 600    CONTINUE
 627    FORMAT(/,1X,'VARIATION OF HORIZONTAL X COMPONENTS IN Y ',
     +           'DIRECTION:',/)
 628    FORMAT(/,1X,'VARIATION OF HORIZONTAL Y COMPONENTS IN Y ',
     +           'DIRECTION:',/)
 629    FORMAT(/,1X,'VARIATION OF VERTICAL (Z) COMPONENTS IN Y ',
     +           'DIRECTION:',/)
 635    FORMAT(//,'-------------------------------------------------',
     +     /,1X,'AMPLITUDES CORRESPONDING TO NODE: ',I5,/,
     +     3X,'(X= ',E10.3,' )        (Z= ',E10.3,' )')
 636    FORMAT(1X,'       Y    ',
     +         1X,' REAL PART ',
     +         1X,'IMAG. PART ',
     +         1X,' MAGNITUDE ',
     +         1X,'   PHASE   ')
 645      FORMAT(5(2X,E10.3))
C
C       Begin loop to IFT displacements for dview
C
        DO 700 I=NSLCT+1,NIFTS
          DO 701 II=1,514
  701     CWIM(II)=(0.,0.)
C
C         Duplicate amplitudes for m's from 0 to NM2P1 (negative m's).
C
          DO 720 IM=NM2P2,NM
            IIM=NCH-IM
  720     CW(IM,I)=CW(IIM,I)
C
          DO 725 IM=1,NM
  725     CWIM(IM)=CW(IM,I)
C
C         Inverse FFT leaving amplitudes complex.
C
          CALL FOUR2(CWIM,NM,1,+1,1)
          WRITE(23,*)IA(IND(5)+I-1),
     +         REAL(A(IND(1)+(IA(IND(5)+I-1))*3-3)),
     +         REAL(A(IND(1)+(IA(IND(5)+I-1))*3-1))
C
          DO 740 IM=1,NM2P1
            Y=(IM-1)*DY
            CWIM(IM)=FACT*CWIM(IM)
            AMPL=SQRT(REAL(CWIM(IM))**2+AIMAG(CWIM(IM))**2)
            PHAS=ATAN2(AIMAG(CWIM(IM)),REAL(CWIM(IM)))*F2
            WRITE(23,645) Y,CWIM(IM),AMPL,PHAS
            IF(INPLOT.EQ.2) GOTO 741
 740      CONTINUE
 741      CONTINUE
 700    CONTINUE
 100  CONTINUE
      CLOSE (15,STATUS='KEEP')
      CLOSE (16,STATUS='KEEP')
      CLOSE (NTAPEO,STATUS='KEEP')
      CLOSE (22,STATUS='KEEP')
      CLOSE (23,STATUS='KEEP')
      CLOSE (25,STATUS='KEEP')
      STOP
      END
C
```

```
C********************* NODE & ELEMENT I/O SUBROUTINE ********************
C
      SUBROUTINE DATAIN(NUMEL,X,ICONN,IS,MAT,NS,IDIR,IPREP,DY,IVAR)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine DATAIN reads in boundary, nodal, and element data.     C
C                                                                       C
C     Subroutine DATAIN is called by : MAIN                             C
C                                                                       C
C     Subroutine DATAIN makes no external calls.                        C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION X(3,NN),IS(3,NN),ICONN(16,NUMEL),MAT(NUMEL)
      DIMENSION IDIR(50),IVAR(50)
      DIMENSION NS(1),IPREP(1)
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /MAX/ XMAX,ZMAX
      COMMON /MESH/ NELX,NELZ
      COMMON /UNIT/ NTAPE,NTAPEO
      IZERO=0
      DO 5 I=1,NN
        IS(2,I)=0.
        DO 5 J=1,3
    5 IS(J,I)=IZERO
      IF(INOUT.EQ.1) WRITE(NTAPEO,50)
   50 FORMAT(/,1X,'NODAL COORDINATES',//,
     2          3X,'NODE',9X,'X',14X,'Y',14X,'Z',8X,'IS1',3X,'IS2',
     3          3X,'IS3',//)
C
      ZMAX=0.
      DO 400 I=1,NNT2
        READ(5,*)K,X(1,I),X(3,I),IS(1,I),IS(3,I)
        X(2,I)=0.
        IF(K.NE.I) THEN
          WRITE(6,402)
          STOP
        ELSE
        ENDIF
        ZMAX=MAX(ZMAX,X(3,I))
  400 CONTINUE
C
C     Duplicate face of nodes at y=-dy
C
      DO 410 I=1,NNT2
        II=I+NNT2
        X(1,II)=X(1,I)
        X(2,II)=-DY
        X(3,II)=X(3,I)
  410 CONTINUE
C
C     Set boundary conditions
C
      DO 420 I=1,NN
        IF(ABS(X(3,I)-ZMAX).LT.0.01) THEN
          IS(1,I)=1
          IS(2,I)=1
          IS(3,I)=1
        ELSE
        ENDIF
  420 CONTINUE
      IF(INPLOT.EQ.1) THEN
        DO 425 I=1,NNT2
          WRITE(23,*)I,X(1,I),X(3,I),IS(1,I),IS(3,I)
  425   CONTINUE
      ENDIF
      IF(INOUT.EQ.1) THEN
        DO 430 I=1,NN
  430   WRITE(NTAPEO,401) I,(X(J,I),J=1,3),(IS(J,I),J=1,3)
```

```
      ELSE
      ENDIF
  401 FORMAT(2X,I5,3(3X,E12.4),3(3X,I3))
  402 FORMAT(//,1X,'ERROR: SUBROUTINE DATAIN',/,
     2            ' NODES NOT IN SEQUENTIAL ORDER',//)
C
C     Nodes of interest (increasing order)
C
      WRITE(NTAPEO,60)
   60 FORMAT(/,1X,'NODES OF INTEREST:',/)
      DO 499 I=1,NSLCT+7
  499 NS(I)=0
      DO 500 I=1,NSLCT
        READ(5,*) NS(I),IDIR(I),IVAR(I)
  500 CONTINUE
      DO 510 I=1,NSLCT,8
  510 WRITE(NTAPEO,501) (NS(J),J=I,I+7)
  501 FORMAT (1X,8I6)
      READ(5,*) NELX,NELZ
C
C     Element Connectivities
C
      IF(INOUT.EQ.1) WRITE(NTAPEO,105)
  105 FORMAT(1X,///,1X,'CONNECTIVITIES',/)
      IF(INOUT.EQ.1) WRITE(NTAPEO,107)
  107 FORMAT(1X,/,30X,'----- ELEMENT NUMBERING -----')
      IF(INOUT.EQ.1) WRITE(NTAPEO,109)
      IF(INOUT.EQ.1) WRITE(NTAPEO,110) (J,J=1,8)
  109 FORMAT(7X,'MAT.')
  110 FORMAT(1X,'ELEM',2X,'TYPE',2X,'NODES:',1X,8I6)
      IF(INOUT.EQ.1) WRITE(NTAPEO,111)
  111 FORMAT(1X,'****',2X,'****',2X,
     2 '*******************************************************',/)
      DO 200 IEL=1,NUMEL
        READ(5,*)K,MAT(IEL),(ICONN(J,K),J=1,8)
        DO 210 J=9,16
  210   ICONN(J,K)=ICONN(J-8,K)+NNT2
        IF(K.NE.IEL) THEN
          WRITE(6,201)
          STOP
        ELSE
        ENDIF
        IF(INOUT.EQ.1)
     2    WRITE(NTAPEO,202) IEL,MAT(IEL),(ICONN(J,IEL),J=1,8)
        IF(INOUT.EQ.1) WRITE(NTAPEO,203) (ICONN(J,IEL),J=9,16)
        IF(INPLOT.EQ.1) THEN
          WRITE(23,202) IEL,MAT(IEL),
     2       ICONN(1,IEL),ICONN(3,IEL),ICONN(8,IEL),ICONN(6,IEL)
        ENDIF
  200 CONTINUE
  201 FORMAT(//,1X,'ERROR: SUBROUTINE DATAIN',/,
     2            ' ELEMENT CONNECTIVITY NOT IN SEQUENCE',/)
  202 FORMAT(1X,I3,3X,I3,10X,8I6)
  203 FORMAT(20X,8I6)
C
C     Old Subroutine PREP
C     Fix to 8 for 16-node element in 2-D
C
      L=0
      DO 120 IEL=1,NUMEL
        DO 115 J=1,NNE2
          L=L+1
          NODE=ICONN(J,IEL)
          IPREP(L)=10*NODE+3
  115   CONTINUE
C
  120 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELP3D(D,RO,G)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine ELP3D is used to read and store material information.  C
C                                                                       C
C     Subroutine ELP3D is called by: MAIN                               C
C                                                                       C
C     Subroutine ELP3D makes no external calls.                         C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION RO(NOMAT),G(NOMAT),MCHK(NOMAT)
      REAL DAMP,PR
      COMPLEX D(3,NOMAT),GC,DAMPC,Z,FACTOR
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /UNIT/ NTAPE,NTAPEO
      Z=(0.,0.)
      WRITE(NTAPEO,100)
      WRITE(NTAPEO,101)
  100 FORMAT(//,'*****MATERIAL PROPERTIES : ',/)
  101 FORMAT(T13,'SHEAR',T28,'POISSONS',T43,'DAMPING',T60,'MASS',/,
     2 T2,'MAT',T12,'MODULUS',T29,'RATIO',T44,'RATIO',T58,'DENSITY',/,
     3 T2,'***',T10,'***********',T28,'*******',T43,'*******',T58,
     4 '*******')
C
      DO 50 I=1,NOMAT
        READ(5,*) M,G(I),PR,DAMP,RO(I)
        MCHK(I)=M
        IF(I.GT.1) THEN
          DO 51 J=1,I-1
            IF(M.EQ.MCHK(J)) THEN
              WRITE(NTAPEO,*)'ERROR IN ELP3D:  MATERIAL NUMBERING'
            ENDIF
   51     CONTINUE
        ENDIF
        WRITE(NTAPEO,102) M,G(I),PR,DAMP,RO(I)
  102   FORMAT(T2,I3,T9,E12.5,T30,F4.2,T45,F4.2,T58,F6.2)
C
C       Choose form of damping
C
        DAMPC=CMPLX(1.,2.*DAMP)
c       DAMPC=CMPLX(1.-2.*DAMP**2.,2.*DAMP*SQRT(1.-DAMP**2.))
        GC=G(I)*DAMPC
        FACTOR=2.*GC/(1.-2.*PR)
        D(1,M)=FACTOR*(1.-PR)
        D(2,M)=FACTOR*PR
        D(3,M)=GC
   50 CONTINUE
      IF(INOUT.EQ.0) WRITE(NTAPEO,*)
      IF(INOUT.EQ.1) THEN
        DO 60 I=1,NOMAT
          WRITE(NTAPEO,190) I
  190     FORMAT(//,1X,'THE D MATRIX FOR MATERIAL ',I3,' IS: ',/)
          WRITE(NTAPEO,210) D(1,I),D(2,I),D(2,I),Z,Z,Z
          WRITE(NTAPEO,210) D(2,I),D(1,I),D(2,I),Z,Z,Z
          WRITE(NTAPEO,210) D(2,I),D(2,I),D(1,I),Z,Z,Z
          WRITE(NTAPEO,210) Z,Z,Z,D(3,I),Z,Z
          WRITE(NTAPEO,210) Z,Z,Z,Z,D(3,I),Z
          WRITE(NTAPEO,210) Z,Z,Z,Z,Z,D(3,I)
  210     FORMAT(1X,6('(',E10.3,' +',E10.3,'i)',2X))
   60   CONTINUE
      ELSE
      ENDIF
      RETURN
      END
C
C
C
```

```
C********************* MATRIX SUBROUTINES ********************************
C
      SUBROUTINE XLOAD(X,ICONN,XL,XR,NUMEL,CPD)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine XLOAD is used to quantify the extent of the distri-    C
C     bution of load in the x-dimension.  The extent is specified       C
C     by endpoints assuming uniform distribution between the            C
C     endpoints.                                                        C
C                                                                       C
C     Nodal forces are calculated ONLY for a 2-D plane, the central     C
C     plane of the mesh (automatic condensation).                       C
C                                                                       C
C     Subroutine XLOAD is called by: MAIN                               C
C                                                                       C
C     Subroutine XLOAD makes no external calls.                         C
C                                                                       C
C     Subroutine checked ok for uniform loads 12/18/90.                 C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION X(3,NN),ICONN(16,NUMEL),PD(NDOFT2)
      COMPLEX CPD(NDOFT2)
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /MESH/ NELX,NELZ
      COMMON /UNIT/ NTAPE,NTAPEO
      NUMEL2=NUMEL/2
      DO 10 I=1,NDOFT2
        CPD(I)=(0.,0.)
   10 PD(I)=0.
C
C     Find the range of elements affected by load in x-direction
C
      NELFT=0
      NELRT=0
      IF(ABS(XL-XR).LT.0.0001) THEN
C
C       Point load (match to closest node)
C
        DO 95 I=1,NELX
          WRITE(22,*)'DO 95, POINT LOAD'
          NDOFCNT=(I-1)*24
          IF(X(1,ICONN(6,I)).GE.XL) THEN
            XRP=X(1,ICONN(6,I))-XL
            XCP=ABS(X(1,ICONN(4,I))-XL)
            XLP=XL-X(1,ICONN(1,I))-XL
            IF(XRP.LE.XCP.AND.XRP.LE.XLP) THEN
              IFIX=ICONN(6,I)
              NELFT=I
              NELRT=I+1
              PD(NDOFCNT+18)=1./2.
              PD(NDOFCNT+27)=1./2.
            ELSEIF(XLP.LE.XCP.AND.XLP.LE.XRP) THEN
              IFIX=ICONN(1,I)
              NELFT=I-1
              NELRT=I
              PD(NDOFCNT+3)=1./2.
              IF(NDOFCNT+3.LE.6) PD(NDOFCNT+3)=1.0
              PD(NDOFCNT-6)=1./2.
            ELSEIF(XCP.LT.XLP.AND.XCP.LT.XRP) THEN
              IFIX=ICONN(4,I)
              NELFT=I
              NELRT=I
              PD(NDOFCNT+12)=1.
            ENDIF
            GOTO 96
          ENDIF
   95   CONTINUE
   96   CONTINUE
```

```
        NELLD=NELRT-NELFT+1
        WRITE(6,*)'pOINT LOAD AT NODE: ',IFIX
        WRITE(NTAPEO,*)'POINT LOAD AT NODE: ',IFIX
      ELSE
C
C       Distributed load
C
        WRITE(22,*)'DISTRIBUTED LOAD'
        WRITE(NTAPEO,*)'DISTRIBUTED LOAD'
        DO 100 I=1,NELX
          WRITE(22,*)'DO 100'
          IF(NELFT.EQ.0.AND.X(1,ICONN(6,I)).GE.XL) NELFT=I
          IF(NELFT.NE.0.AND.X(1,ICONN(6,I)).GE.XR) THEN
            NELRT=I
            GOTO 101
          ELSE
          ENDIF
  100   CONTINUE
  101   CONTINUE
        NELLD=NELRT-NELFT+1
C
C     Calculate nodal forces
C
        ICOUNT=NELFT-1
        XEND=XL
C
        DO 200 I=ICOUNT+1,ICOUNT+NELLD
          NDOFCNT=(I-1)*24
          EW=X(1,ICONN(6,I))-X(1,ICONN(1,I))
          EWL=X(1,ICONN(4,I))-X(1,ICONN(1,I))
          EWR=X(1,ICONN(6,I))-X(1,ICONN(4,I))
          WRITE(22,*)'EW=',ew
          WRITE(22,*)'EWL=',EWL
          WRITE(22,*)'EWR=',EWR
          IF(I.EQ.ICOUNT+1.OR.I.EQ.ICOUNT+NELLD) THEN
C
C         End element(s)
C
            WRITE(22,*)'  END ELEMENT'
            IF(X(1,ICONN(4,I)).EQ.XEND) THEN
            WRITE(22,*)'    center node'
C
C           At center node
C
              ETA=0.
            ELSEIF(XEND.LT.X(1,ICONN(4,I))) THEN
            WRITE(22,*)'    left segment'
C
C           In left segment
C
              IF(NELLD.EQ.1)XEND=XR
              ETA=(XEND-X(1,ICONN(4,I)))/
     2                (X(1,ICONN(4,I))-X(1,ICONN(1,I)))
            ELSE
            WRITE(22,*)'    right segment'
C
C           In right segment
C
              ETA=(XEND-X(1,ICONN(4,I)))/
     2                (X(1,ICONN(6,I))-X(1,ICONN(4,I)))
            ENDIF
            WRITE(22,*)'      ETA=',eta
            IF(NELLD.EQ.1) XEND=XR
            IF(ABS(XR-XEND).LT.0.1.OR.(NELLD.EQ.1.AND.ETA.LT.0.)) THEN
              WRITE(22,*)'      forces for right end element'
C
C           Forces for right end element
C
```

```
            IF(ETA.LT.0.)EWR=EWL
            PD(NDOFCNT+3)=((1./6.*ETA**3.)-(0.25*ETA**2.))*EWR
     2                   +5./12.*EWL
            PD(NDOFCNT+12)=(ETA-(1./3.*ETA**3.))*EWR+2./3.*EWL
            PD(NDOFCNT+18)=((1./6.*ETA**3.)+(0.25*ETA**2.))*EWR
     2                   -1./12.*EWL
          ELSEIF(ABS(XL-XEND).LT.0.1.OR.(NELLD.EQ.1.AND.ETA.GT.0.))
     2      THEN
            WRITE(22,*)'      forces for left end element'
C
C         Forces for left end element
C
            IF(ETA.GT.0.)EWL=EWR
            PD(NDOFCNT+3)=-((1./6.*ETA**3.)-(0.25*ETA**2.))*EWL
     2                   -1./12.*EWR
            PD(NDOFCNT+12)=-(ETA-(1./3.*ETA**3.))*EWL+2./3.*EWR
            PD(NDOFCNT+18)=-((1./6.*ETA**3.)+(0.25*ETA**2.))*EWL
     2                   +5./12.*EWR
          ENDIF
          XEND=XR
        ELSE
            WRITE(22,*)'      forces for center element'
C
C       Center element
C
            PD(NDOFCNT+3)=-1./12.*EWR+5./12.*EWL
            PD(NDOFCNT+12)=2./3.*EWR+2./3.*EWL
            PD(NDOFCNT+18)=5./12.*EWR-1./12.*EWL
          ENDIF
  200   CONTINUE
  201   CONTINUE
      ENDIF
C
C     Write load vector to fort.22 for optional inspection
C
      IF(INOUT.EQ.1) THEN
        WRITE(NTAPEO,*)
        WRITE(NTAPEO,*)'NELFT= ',NELFT
        WRITE(NTAPEO,*)'NELRT= ',NELRT
        WRITE(NTAPEO,*)
        WRITE(NTAPEO,*)'Number of elements in x-direction ',
     2                 'affected by load: ',nelld
      ENDIF
      WRITE(22,301)
      DO 300 I=1,NDOFT2/3
        IF(ABS(PD(3*I-2)).GT.1.E-5.OR.ABS(PD(3*I-1)).GT.1.E-5.OR.
     2     ABS(PD(3*I)).GT.1.E-5)
     3     WRITE(22,302) INT(FLOAT(I)/FLOAT(NNE2))+1,
     4          INT(FLOAT(MOD(I,NNE2))),PD(3*I-2),PD(3*I-1),PD(3*I)
  300 CONTINUE
  301 FORMAT(/,1X,'Non-zero nodal forces (from XLOAD):',//,
     2          1X,'  ELEM ','  NODE',13X,'X',16X,'Y',16X,'Z',/)
  302 FORMAT(1X,I6,1X,I6,3(5X,F12.5))
  399 CONTINUE
      DO 400 I=1,NDOFT2
  400 CPD(I)=CMPLX(PD(I))
      RETURN
      END
C
C
C
```

```
      SUBROUTINE YLOAD(CFOR,DY,NM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine YLOAD is used to quantify the extent of the distri-    C
C     bution of load in the y-dimension.  The load is assumed to be     C
C     symmetric (centered about the x-axis).                            C
C                                                                       C
C     At present, the load is assumed to be                             C
C     constant over a distance YLDIS from the line of symmetry.         C
C                                                                       C
C     Subroutine YLOAD is called by: MAIN                               C
C                                                                       C
C     Subroutine YLOAD makes no external calls.                         C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION PM(NM)
      COMPLEX CFOR(NM+2)
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /UNIT/ NTAPE,NTAPEO
      READ(5,*)PMAG,YLDIS
      YTOT=DY*FLOAT(NM)
      WRITE(25,*)YLDIS
C
C     Define distribution of loads is y-direction.
C     At present, uniform load; load duplicated at end of y-space
C
      IF(YLDIS.LE.0.0001) THEN
        DO 50 I=1,NM+2
  50    PM(I)=0.
        PM(1)=PMAG
        DO 60 I=1,NM+2
  60    CFOR(I)=CMPLX(PM(I))
      ELSE
        ICOUNT=0
        DO 100 I=1,NM+2
          IF((I-1)*DY.LE.YLDIS) THEN
            ICOUNT=ICOUNT+1
            PM(I)=PMAG
          ELSE
            PM(I)=0.
          ENDIF
 100    CONTINUE
        DO 200 I=1,ICOUNT-1
          J=NM+1-I
          PM(J)=PMAG
 200    CONTINUE
        DO 300 I=1,NM
C
C     THE FOLLOWING LINE IS USED ONLY WHEN CONSIDERING
C     NORMALIZED LOAD (TOTAL LOAD = 1)!!
C
          PM(I)=PM(I)/(2.*(FLOAT(ICOUNT-1))+1.)
 300    CFOR(I)=CMPLX(PM(I))
        CFOR(NM+1)=(0.,0.)
      ENDIF
      WRITE(NTAPEO,301)
 301 FORMAT(//,'*****LOADS:',//,
     1' SUBROUTINE YLOAD ASSUMES THAT A NORMALIZED LOAD IS BEING USED!',
     2 /)
      RETURN
      END
C
C
C
```

```
      SUBROUTINE STIFF(X,IS,ICONN,MAT,D,ELRO,ELG,NUMEL,OM,E1,E2,IM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                   C
C     SUBROUTINE STIFF FORMS THE STIFFNESS MATRIX AND ADJUSTS THE RHS C
C     VECTOR TO ACCOUNT FOR BOUNDARY CONDITIONS.                      C
C                                                                   C
C     Since elements adjacent in the y-direction have the same stiff- C
C     ness, the stiffness matrices for one front set are calculated.  C
C     Then in CONDENSE, the stiffness matrices of the two elements    C
C     adjacent in the y-direction are combined.                       C
C                                                                   C
C     Subroutine STIFF is called by : MAIN                            C
C                                                                   C
C     Subroutine STIFF calls: MODIF, ISO16, & CONDENSE                C
C                                                                   C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION X(3,NN),Y(3,16),ICONN(16,NUMEL),MAT(NUMEL),IS(3,NN)
      DIMENSION ELRO(NOMAT),ELG(NOMAT),RHS(48)
      COMPLEX D(3,NOMAT),DD(6,6),CSM(48,48),CSMC(NDOFE2,NDOFE2),E1,E2
      COMPLEX CPDEL(NDOFE2)
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /UNIT/ NTAPE,NTAPEO
      REWIND 10
      REWIND 12
      NUMEL2=NUMEL
      DO 10 IEL=1,NUMEL2
C
C       PICK coordinates for element nodes from X array
C
        DO 15 J=1,16
          NODE=ICONN(J,IEL)
          DO 15 K=1,3
            Y(K,J)=X(K,NODE)
   15   CONTINUE
C
C       DD is material array for specific element
C
        DO 305 KK=1,6
          DO 305 LL=1,6
  305   DD(KK,LL)=(0.0,0.0)
C
        DD(1,1)=D(1,MAT(IEL))
        DD(1,2)=D(2,MAT(IEL))
        DD(1,3)=D(2,MAT(IEL))
        DD(2,1)=D(2,MAT(IEL))
        DD(2,2)=D(1,MAT(IEL))
        DD(2,3)=D(2,MAT(IEL))
        DD(3,1)=D(2,MAT(IEL))
        DD(3,2)=D(2,MAT(IEL))
        DD(3,3)=D(1,MAT(IEL))
        DD(4,4)=D(3,MAT(IEL))
        DD(5,5)=D(3,MAT(IEL))
        DD(6,6)=D(3,MAT(IEL))
C
        FAC=ELRO(MAT(IEL))*OM*OM
        CALL ISO16(Y,DD,CSM,RHS,FAC)
        IF(IM.LE.1)
     2         CALL PRNTRHS(CPDEL,24,IEL,IM,4,ELG(MAT(IEL)))
        DO 301 II=1,NDOFE2
          READ(12) CPDEL(II)
  301   CONTINUE
C
        CALL MODIF(CSM,IS,CPDEL,ICONN(1,IEL),IM)
        CALL SYMSM(DUM,CSM,48,IEL,ELG(MAT(IEL)),1,5)
        CALL CONDENSE(CSM,CSMC,E1,E2,IM,IEL)
        IF(IM.LE.1)
     2         CALL PRNTRHS(CPDEL,24,IEL,IM,6,ELG(MAT(IEL)))
```

```
      IF(IEL.EQ.2.AND.IM.LE.3)
     2         CALL PRNTSM(CSMC,24,IEL,IM,6,ELG(MAT(IEL)))
      CALL SYMSM(DUM,CSMC,24,IEL,ELG(MAT(IEL)),1,6)
      WRITE(10)((CSMC(I,J),I=1,J),J=1,NDOFE2),(CPDEL(I),I=1,NDOFE2)
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE MODIF(CSM,IS,CPDEL,ICONN,IM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine MODIF is used to modify the stiffness matrix and     C
C     RHS vector to account for boundary conditions.  The matrix      C
C     is then checked for symmetry.  The matrix can be printed if     C
C     desired.  The RHS vector is changed from real to complex.       C
C                                                                     C
C     Subroutine MODIF is called by: STIFF                            C
C                                                                     C
C     Subroutine MODIF calls: PRNTSM (optional)                       C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION IS(3,NN),ICONN(16)
      COMPLEX CSM(48,48),CPDEL(24)
      COMMON /BIG/ BIG
      COMMON /INFO/ NN,NOMAT,NF,NSLCT,INOUT,INPLOT
      COMMON /MAX/ XMAX,ZMAX
      COMMON /UNIT/ NTAPE,NTAPEO
      K=0
C
C     ICONN is with respect to an element
C
      DO 10 I=1,16
        NODE=ICONN(I)
        DO 10 J=1,3
          K=K+1
          IF(IS(J,NODE).EQ.1) THEN
            CSM(K,K)=CSM(K,K)+BIG
          ELSEIF(IS(J,NODE).NE.0) THEN
            WRITE(6,20)
            STOP
          ELSE
          ENDIF
   20 FORMAT(//,1X,'***ERROR IN MODIF (with IS)***',//)
   10 CONTINUE
C
C     No (static) forces
C     Zero body forces
C
      K=0
      DO 25  I=1,8
        NODE=ICONN(I)
        DO 25 J=1,3
          K=K+1
          IF(IS(J,NODE).EQ.1) CPDEL(K)=CPDEL(K)*BIG
   25 CONTINUE
      RETURN
      END
C
C
C
```

```
      SUBROUTINE CONDENSE(CSM,CSMC,E1,E2,IM,iel)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                         C
C     Subroutine CONDENSE is used to derive an equivalent 2-D             C
C     stiffness matrix from a three-dimensional stiffness matrix.         C
C     A 16-node isoparametric element quadratic in the y and z            C
C     directions and linear in the x direction is assumed.                C
C                                                                         C
C     The RHS vector does not require condensation because only           C
C     loads on the central plane are used.                                C
C                                                                         C
C     Subroutine CONDENSE is called by STIFF.                             C
C                                                                         C
C     Subroutine CONDENSE makes no external calls.                        C
C                                                                         C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /TOL/ TOLSM
      COMPLEX CSM(48,48),CSMC(NDOFE2,NDOFE2)
      COMPLEX E1,E2,AI
      DO 10 I=1,NDOFE2
        DO 10 J=1,NDOFE2
   10 CSMC(I,J)=(0.,0.)
C
      DO 20 I=1,NDOFE2
        II=I+NDOFE2
        DO 20 J=1,NDOFE2
          JJ=J+NDOFE2
          CSMC(I,J)=CSM(II,J)*E2+CSM(I,J)+CSM(II,JJ)+CSM(I,JJ)*E1
   20 CONTINUE
C
C     Make matrix symmetric through division and multiplication by i.
C
      IF(IM.EQ.1) GOTO 31
      AI=(0.,1.)
      II=1
      DO 40 I=1,NDOFE2
        JJ=1
        DO 30 J=1,NDOFE2
          IF(I.EQ.II*3-1) CSMC(I,J)=CSMC(I,J)*AI
          IF(J.EQ.JJ*3-1) THEN
            JJ=JJ+1
            CSMC(I,J)=CSMC(I,J)/AI
          ENDIF
  30    CONTINUE
        IF(I.EQ.II*3-1) II=II+1
  40  CONTINUE
  31  CONTINUE
      RETURN
      END
C
C
C
```

```
C********************* ELEMENT SUBROUTINES ******************************
C
      SUBROUTINE ISO16(Y,D,CSM,RHS,FAC)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine ISO16 calculates the stiffness matrix for a 3-D,       C
C     16-node isoparametric finite element.                             C
C                                                                       C
C     Subroutine ISO16 was validated (ISO16S) along with DNISO16Y in    C
C     gen_fem.                                                          C
C                                                                       C
C     Subroutine ISO16 is called by: MAIN.                              C
C                                                                       C
C     Subroutine ISO16 calls subroutines: DNISO16                       C
C                                                                       C
C     Variable list:                                                    C
C                                                                       C
C        AJACS          Volume of element (det. of Jacobian)          C
C        AN             Interpolation functions matrix                C
C        ANT            Transpose of interpolation functions matrix   C
C        AUX            Temporary matrix of AUXC                      C
C        AUXB           Matrices B transpose and D multiplied         C
C        AUXC           Matrices B transpose,D, and B multiplied      C
C        AUXR           Consistent mass matrix                        C
C        B                                                            C
C        BT             Transpose of ??? matrix                       C
C        CSM            Complex stiffness matrix (additive)           C
C        D              Material property matrix                      C
C        DNDXI          Matrix of deriv. of interpolation functions   C
C                            with respect to XI                       C
C        RHS            Matrix of body forces (zero)                  C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /INTERPF/ AN1D(16,18),DNDXI(3,16,18),W(18)
      REAL Y(3,16),B(6,48),BT(48,6),RHS(48),DXX(3,3),DXI(3,3),DNDX(3)
      REAL AN(3,48),ANT(48,3),AUXR(48,48)
      INTEGER LOOP3(9)
      DATA LOOP3 /1,2,3,1,2,3,1,2,3/
      COMPLEX CSM(48,48),D(6,6),AUXB(48,6),AUXC(48,48),AUX(48)
      COMPLEX CSUM
C
C     Zero arrays
C       RHS=0:  Zero body forces
C
      DO 10 I=1,48
        AUX(I)=(0.,0.)
        RHS(I)=0.
        DO 20 J=1,6
          AUXB(I,J)=(0.,0.)
          B(J,I)=0.
   20   BT(I,J)=0.
        DO 25 J=1,48
          AUXR(I,J)=0.
          AUXC(I,J)=(0.,0.)
   25   CSM(I,J)=(0.,0.)
   10 CONTINUE
      DO 30 I=1,3
        DNDX(I)=0.
        DO 30 J=1,3
   30 DXI(I,J)=0.
C
C     Loop on integration points to create element stiff. matrix
C
      DO 1000 IP=1,18
        DO 39 I=1,3
          DO 39 J=1,48
            AN(I,J)=0.0
            ANT(J,I)=0.0
   39   CONTINUE
```

```
      DO 40 I=1,16
        K=3*I
        AN(3,K)=AN1D(I,IP)
        AN(2,K-1)=AN1D(I,IP)
        AN(1,K-2)=AN1D(I,IP)
   40 CONTINUE
      DO 50 I=1,3
        DO 50 J=1,3
   50 DXX(I,J)=0.
C
C     Perform a summation to get the components of the
C     Jacobian Matrix
C
      DO 100 INODE=1,16
        DO 100 II=1,3
          DO 100 JJ=1,3
  100 DXX(JJ,II)=DXX(JJ,II)+DNDXI(II,INODE,IP)*Y(JJ,INODE)
C
C     Calculate determinant of DXX
C
      SUMP=0.
      SUMM=0.
      K=-1
      DO 120 II=1,3
        K=K+2
  120 SUMP=SUMP+DXX(LOOP3(II),LOOP3(K))*DXX(LOOP3(II+1),LOOP3(K+1))
     2                                  *DXX(LOOP3(II+2),LOOP3(K+2))
      DO 130 II=1,3
  130 SUMM=SUMM+DXX(LOOP3(II),3)*DXX(LOOP3(II+1),2)
     2                           *DXX(LOOP3(II+2),1)
      DET=SUMP-SUMM
      AJACS=ABS(DET)*W(IP)
C
C     Calculate the inverse of the Jacobian Matrix
C
      DO 140 II=1,3
        DO 140 JJ=1,3
          FAC2=1.
          IF(JJ.EQ.2.OR.II.EQ.2) FAC2=-1.
          IF(JJ.EQ.2.AND.II.EQ.2) FAC2=1.
  140 DXI(JJ,II)=(FAC2/DET)*(-1.)**(II+JJ)
     2     *(DXX(LOOP3(II+1),LOOP3(JJ+1))*DXX(LOOP3(II+2),LOOP3(JJ+2))
     3     -DXX(LOOP3(II+1),LOOP3(JJ+2))*DXX(LOOP3(II+2),LOOP3(JJ+1)))
C
C     Loop on nodes to calculate B matrix
C
      DO 195 KK=1,3
  195 DNDX(KK)=0.
      DO 200 INODE=1,16
        DO 210 II=1,3
          SUM=0.
          DO 210 JJ=1,3
            SUM=SUM+DXI(JJ,II)*DNDXI(JJ,INODE,IP)
  210     DNDX(II)=SUM
          J1=3*INODE-2
          J2=3*INODE-1
          J3=3*INODE
          B(1,J1)=DNDX(1)
          B(4,J1)=DNDX(2)
          B(6,J1)=DNDX(3)
          B(2,J2)=DNDX(2)
          B(4,J2)=DNDX(1)
          B(5,J2)=DNDX(3)
          B(3,J3)=DNDX(3)
          B(5,J3)=DNDX(2)
          B(6,J3)=DNDX(1)
  200 CONTINUE
C
```

```
      DO 55 M=1,3
        DO 55 L=1,48
   55   ANT(L,M)=AN(M,L)
C
      DO 60 M=1,6
        DO 60 L=1,48
   60   BT(L,M)=B(M,L)
C
        DO 70 L=1,48
          DO 70 M=1,48
            SUM=0.
            DO 71 N=1,3
   71         SUM=SUM+ANT(L,N)*AN(N,M)
   70     AUXR(L,M)=SUM
C
        DO 80 L=1,48
          DO 80 M=1,6
            CSUM=(0.,0.)
            DO 81 N=1,6
   81       CSUM=CSUM+BT(L,N)*D(N,M)
   80     AUXB(L,M)=CSUM
C
        DO 90 L=1,48
          DO 91 M=1,48
            CSUM=(0.,0.)
            DO 92 N=1,6
   92       CSUM=CSUM+AUXB(L,N)*B(N,M)
   91       AUX(M)=CSUM
            DO 90 M=1,48
   90     AUXC(L,M)=AUX(M)
C
        DO 110 L=1,48
          DO 110 M=1,48
  110     CSM(L,M)=CSM(L,M)+AJACS*(AUXC(L,M)-FAC*AUXR(L,M))
 1000   CONTINUE
        RETURN
        END
C
C
C
C********************* NUMERICAL INTERPOLATION SUBROUTINES ********************
C
C
      SUBROUTINE DNISO16YF
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     This subroutine calculates the values of interpolation functions          C
C     and differentiated interpolation functions at numerical integration       C
C     points.                                                                   C
C     This subroutine uses Gauss weight factors for 3 x 3 x 2 integration       C
C                                                                               C
C     This routine assumes the following orientation:                           C
C                                                                               C
C     ----------->  + X                                                         C
C     |                                                                         C
C     |           1 --- 4 --- 6                 9 -- 12 -- 14                   C
C     |           |           |                 |           |                   C
C     |           |           |                 |           |                   C
C    \ /          2   (+Y)    7       and       10   (-Y)   15                  C
C     .           |           |                 |           |                   C
C    + Z          3 --- 5 --- 8                 11 -- 13 -- 16                  C
C                                                                               C
C                                                                               C
C     INTEGRATION POINTS:                                                       C
C                                                                               C
C                                                                               C
```

```
C     --------> +X                                                          C
C     |                                                                     C
C     |          ----------------                    --------------------   C
C     |          |              |                    |                  |   C
C     |          | x 1  x 4  x 7|                    | x 10  x 13  x 16 |   C
C     \ /        | x 2  x 5  x 8|        AND         | x 11  x 14  x 17 |   C
C      .         | x 3  x 6  x 9|                    | x 12  x 15  x 18 |   C
C     +Z         |              |                    |                  |   C
C                ----------------                    --------------------   C
C                                                                           C
C                                                                           C
C     Subroutine DNISO16YF may be called by: ISO16                          C
C                                                                           C
C     Subroutine DNISO16YF makes no reference calls.                        C
C                                                                           C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /INTERPF/ AN1D(16,18),DNDXI(3,16,18),W(18)
      A=5./9.
      B=8./9.
      W(1)=A*A
      W(3)=W(1)
      W(7)=W(1)
      W(9)=W(1)
      W(2)=A*B
      W(4)=A*B
      W(6)=A*B
      W(8)=A*B
      W(5)=B*B
      DO 10 I=10,18
   10 W(I)=W(I-9)
C
      L=0
C
C     Loop on Y (s) from positive to negative
C
      STEMP=1./SQRT(3.)
      RTEMP=SQRT(3.)/SQRT(5.)
      TTEMP=SQRT(3.)/SQRT(5.)
      DO 100 I=1,2
        S=STEMP-FLOAT(I-1)*2.0*STEMP
C
C       Loop on X (r) from negative to positive
C
        R=-2.*RTEMP
        DO 200 J=1,3
          R=R+RTEMP
C
C         Loop on Z (t) from negative to positive
C
          T=-2.*TTEMP
          DO 300 K=1,3
            T=T+TTEMP
C
            L=L+1
            R2=R*R
            T2=T*T
            RT=R*T
C
            AN1D(1,L)=0.125*(1.+S)*(-1.+RT+R2*(1.-T)+T2*(1.-R))
            AN1D(2,L)=0.25*(1.-R)*(1.+S)*(1.-T2)
            AN1D(3,L)=0.125*(1.+S)*(-1.-RT+R2*(1.+T)+T2*(1.-R))
            AN1D(4,L)=0.25*(1.-R2)*(1.+S)*(1.-T)
            AN1D(5,L)=0.25*(1.-R2)*(1.+S)*(1.+T)
            AN1D(6,L)=0.125*(1.+S)*(-1.-RT+R2*(1.-T)+T2*(1.+R))
            AN1D(7,L)=0.25*(1.+R)*(1.+S)*(1.-T2)
            AN1D(8,L)=0.125*(1.+S)*(-1.+RT+R2*(1.+T)+T2*(1.+R))
            AN1D(9,L)=0.125*(1.-S)*(-1.+RT+R2*(1.-T)+T2*(1.-R))
            AN1D(10,L)=0.25*(1.-R)*(1.-S)*(1.-T2)
            AN1D(11,L)=0.125*(1.-S)*(-1.-RT+R2*(1.+T)+T2*(1.-R))
```

```
            AN1D(12,L)=0.25*(1.-R2)*(1.-S)*(1.-T)
            AN1D(13,L)=0.25*(1.-R2)*(1.-S)*(1.+T)
            AN1D(14,L)=0.125*(1.-S)*(-1.-RT+R2*(1.-T)+T2*(1.+R))
            AN1D(15,L)=0.25*(1.+R)*(1.-S)*(1.-T2)
            AN1D(16,L)=0.125*(1.-S)*(-1.+RT+R2*(1.+T)+T2*(1.+R))
C
            DNDXI(1,1,L)=0.125*(2.*R+T)*(1.-T)*(1.+S)
            DNDXI(1,2,L)=-0.25*(1.+S)*(1.-T2)
            DNDXI(1,3,L)=0.125*(2.*R-T)*(1.+S)*(1.+T)
            DNDXI(1,4,L)=-R/2.*(1.+S)*(1.-T)
            DNDXI(1,5,L)=-R/2.*(1.+S)*(1.+T)
            DNDXI(1,6,L)=0.125*(2.*R-T)*(1.+S)*(1.-T)
            DNDXI(1,7,L)=0.25*(1.+S)*(1.-T2)
            DNDXI(1,8,L)=0.125*(2.*R+T)*(1.+S)*(1.+T)
            DNDXI(1,9,L)=0.125*(2.*R+T)*(1.-S)*(1.-T)
            DNDXI(1,10,L)=-0.25*(1.-S)*(1.-T2)
            DNDXI(1,11,L)=0.125*(2.*R-T)*(1.-S)*(1.+T)
            DNDXI(1,12,L)=-R/2.*(1.-S)*(1.-T)
            DNDXI(1,13,L)=-R/2.*(1.-S)*(1.+T)
            DNDXI(1,14,L)=0.125*(2.*R-T)*(1.-S)*(1.-T)
            DNDXI(1,15,L)=0.25*(1.-S)*(1.-T2)
            DNDXI(1,16,L)=0.125*(2.*R+T)*(1.-S)*(1.+T)
            DNDXI(2,1,L)=0.125*(-1.+RT+R2*(1.-T)+T2*(1.-R))
            DNDXI(2,2,L)=0.25*(1.-R)*(1.-T2)
            DNDXI(2,3,L)=0.125*(-1.0-RT+R2*(1.+T)+T2*(1.-R))
            DNDXI(2,4,L)=0.25*(1.-R2)*(1.-T)
            DNDXI(2,5,L)=0.25*(1.+T)*(1.-R2)
            DNDXI(2,6,L)=0.125*(-(1.+RT)+R2*(1.-T)+T2*(1.+R))
            DNDXI(2,7,L)=0.25*(1.+R)*(1.-T2)
            DNDXI(2,8,L)=0.125*(-1.+RT+R2*(1.+T)+T2*(1.+R))
            DNDXI(2,9,L)=0.125*(1.-RT-R2*(1.-T)-T2*(1.-R))
            DNDXI(2,10,L)=-0.25*(1.-R)*(1.-T2)
            DNDXI(2,11,L)=0.125*(1.+RT-R2*(1.+T)-T2*(1.-R))
            DNDXI(2,12,L)=-0.25*(1.-T)*(1.-R2)
            DNDXI(2,13,L)=-0.25*(1.+T)*(1.-R2)
            DNDXI(2,14,L)=0.125*(1.+RT-R2*(1.-T)-T2*(1.+R))
            DNDXI(2,15,L)=-0.25*(1.+R)*(1.-T2)
            DNDXI(2,16,L)=0.125*(1.-RT-R2*(1.+T)-T2*(1.+R))
            DNDXI(3,1,L)=0.125*(2.*T+R)*(1.+S)*(1.-R)
            DNDXI(3,2,L)=-T/2.*(1.-R)*(1.+S)
            DNDXI(3,3,L)=0.125*(2.*T-R)*(1.+S)*(1.-R)
            DNDXI(3,4,L)=-0.25*(1.-R2)*(1.+S)
            DNDXI(3,5,L)=0.25*(1.+S)*(1.-R2)
            DNDXI(3,6,L)=0.125*(2.*T-R)*(1.+R)*(1.+S)
            DNDXI(3,7,L)=-T/2.*(1.+R)*(1.+S)
            DNDXI(3,8,L)=0.125*(2.*T+R)*(1.+R)*(1.+S)
            DNDXI(3,9,L)=0.125*(2.*T+R)*(1.-R)*(1.-S)
            DNDXI(3,10,L)=-T/2.*(1.-R)*(1.-S)
            DNDXI(3,11,L)=0.125*(2.*T-R)*(1.-R)*(1.-S)
            DNDXI(3,12,L)=-0.25*(1.-S)*(1.-R2)
            DNDXI(3,13,L)=0.25*(1.-S)*(1.-R2)
            DNDXI(3,14,L)=0.125*(2*T-R)*(1.+R)*(1.-S)
            DNDXI(3,15,L)=-T/2.*(1.+R)*(1.-S)
            DNDXI(3,16,L)=0.125*(2.*T+R)*(1.+R)*(1.-S)
  300     CONTINUE
  200   CONTINUE
  100 CONTINUE
      RETURN
      END
C
C
C
```

```
C********************* MISCELLANEOUS SUBROUTINES ************************
C
      SUBROUTINE DESVEC(MFW,NIN,IPREP,IB,IC)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine DESVEC calculates the destination vectors from         C
C     nicknames.  Arguments MFW, NIN, and array IPREP (defined          C
C     in old PREP now contained in DATAIN) come in.  The length of      C
C     IPREP (IND(11)) is long enough for IB and IC.                     C
C                                                                       C
C     Subroutine DESVEC is called by: PRFNT                             C
C                                                                       C
C     Subroutine DESVEC calls PREOUT.                                   C
C                                                                       C
C     All arguments coming in.                                          C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ ISYM,NUMEL2,IDUM(24),MDOF,MFW,MLDEST
      DIMENSION IPREP(1),IB(1),IC(1)
C
C     The following statement appears to be a function.
C     The calculation is not necessary for vib3 because
C     the MDOF is always 3.
C
      MODR(I,J) = I-I/J*J
      MDOF = 0
      MFW = 0
      IDES = 1
      IP = 0
      JDN = 0
      DO 10 I=1,NFN
   10 IB(I) = 0
      DO 100 IEL=1,NUMEL2
        N = NIN
        NT = 0
        IPS = IP
        IPC = 1
        NE = 0
        NTT = 0
        DO 60 ID=1,N
          IP = IP+1
          INIC = IPREP(IP)
          NDOF = MODR(INIC,10)
          IF(NDOF.NE.3) THEN
            WRITE(6,*)'PROBLEM WITH NDOF IN DESVEC (.NE.3)'
            STOP
          ELSE
          ENDIF
          NT = NT+NDOF
          IF(IB(IP) .GT. 0) GO TO 20
          JDES = IDES
          IB(IP) = IDES*100+NDOF*10
          IDES = IDES+NDOF
          IF(IDES-1 .GT. MFW) MFW = IDES-1
          GO TO 30
   20     JDES = IB(IP)
          IB(IP) = IB(IP)*100+NDOF*10
   30     JP = IPS+N+1
          IF(JP .GT. NFN) GO TO 45
          DO 40 JD=JP,NFN
            IF(INIC .EQ. IPREP(JD)) GO TO 50
   40     CONTINUE
   45     IB(IP) = IB(IP)+1
          IC(IPC) = JDES
          IC(IPC+1) = NDOF
          IPC = IPC+2
          NE = NE+1
          NTT = NTT+NDOF
          GO TO 60
```

```
   50     IB(JD) = JDES
          IF(JD .GT. JDN) JDN=JD
   60   CONTINUE
        IF(NT .GT. MDOF) MDOF = NT
        IF(IEL .EQ. NUMEL2 .OR. NE .EQ. 0) GO TO 90
        IDES = IDES-NTT
        JP = IPS+N+1
        IF(JP .GT. JDN) GO TO 90
        DO 80 JD=JP,JDN
          IF(IB(JD) .EQ. 0) GO TO 80
          IPC = 1
          NT = 0
          DO 70 I=1,NE
            IF(IB(JD) .LT. IC(IPC)) GO TO 70
            NT = NT+IC(IPC+1)
   70     IPC = IPC+2
          IB(JD) = IB(JD)-NT
   80   CONTINUE
   90   CALL PREOUT(IEL,N,IPREP(IPS+1),IB(IPS+1))
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PREFNT(NIN,IA,MS,MU,MR)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                                     C
C     Subroutine PREFNT initiates prefront.                                           C
C     Since SOLVE is in a DO loop, PREFNT also zeros some parameters.                 C
C                                                                                     C
C     Subroutine PREFNT is called by: MAIN                                            C
C                                                                                     C
C     Subroutine PREFNT calls: SECOND & DESVEC                                        C
C                                                                                     C
C     NIN and IA coming in; MS, MU, and MR going out                                  C
C                                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON/CNTL/ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,MA,
     .            IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .            IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .            MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION IA(1)
      CALL SECOND(T0)
      MLDEST=NIN
      NFN=NUMEL2*NIN
      CALL DESVEC(NFN,NIN,IA,IA(NFN+1),IA(2*NFN+1))
      MR = MDOF+MFW+1
      MS = NUMEL2+MLDEST+2*MDOF+(MDOF*(MDOF+1))/2+(MFW*(MFW+1))/2+MFW
      MU = NUMEL2+MLDEST+2*MDOF+MDOF*MDOF+MFW*MFW+MFW
      IWRT=0
      IB=0
      MBUF=0
      MB=0
      IU=0
      MW=0
      IERR=0
      IL=0
      MKF=0
      NNEGP=0
      IFB=0
      MELEM=0
      NPOSP=0
      IFU=0
      MFWR=0
      NRHSF=0
      IFL=0
      CALL SECOND(T1)
      RETURN
      END
```

```
      SUBROUTINE PREOUT(IEL,N,IA,IB)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine PREOUT is used to write destination vectors.                   C
C                                                                               C
C     Subroutine PREOUT is called by: DESVEC                                    C
C                                                                               C
C     Subroutine PREOUT makes no external calls.                                C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION IA(1),IB(1)
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /WORK/ IDEST(1)
      COMMON /FLAGS/ IND(22)
      J=IND(12)+NNE2*(IEL-1)-1
      DO 10 I=1,N
        J=J+1
        IDEST(J)=IB(I)
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE PRNTRHS(CRHS,NN,IEL,IM,ICODE)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine PRNTRHS is an optional routine to print the RHS                C
C     matrix (48 x 1) or (16 x 3).   For default, PRNTRHS prints                C
C     the RHS matrix of element 1 to file (TAPE22).                             C
C                                                                               C
C     Subroutine PRNTRHS is called by: MODIF                                    C
C                                                                               C
C     Subroutine PRNTRHS makes no external calls.                               C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION RHS(NN)
      COMPLEX CRHS(NN)
      DO 10 I=1,NN
   10 RHS(I)=SQRT(REAL(CRHS(I))**2.+AIMAG(CRHS(I))**2.)
      WRITE(22,*)
      IF(ICODE.EQ.4)WRITE(22,*)'AFTER ISO16:'
      IF(ICODE.EQ.5)WRITE(22,*)'AFTER MODIF:'
      IF(ICODE.EQ.6)WRITE(22,*)'AFTER CONDENSE:'
      IF(ICODE.EQ.7)WRITE(22,*)'AFTER SOLVE:'
      WRITE(22,15)IEL,IM-1
   15 FORMAT(1X,'MODULUS OF COMPLEX RHS VECTOR- ',/,
     1       1X,'ELEMENT: ',I5,'  WAVENUMBER: ',I3,/,
     2'  NODE          X               Y               Z',/)
      DO 25 I=1,NN/3
   25 WRITE(22,20)I,RHS(3*I-2),RHS(3*I-1),CRHS(3*I)
   20 FORMAT(1X,I5,5X,2(E11.4,4X),2X,E11.4,' + ',E11.4,' i')
      RETURN
      END
C
C
C
      SUBROUTINE PRNTSM(CSM,NN,IEL,IM,ICODE,G)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine PRNTSM is an optional routine to print the stiffness           C
C     matrix (NN x NN).   At default, PRNTSM prints the stiffness               C
C     matrix of element 1 to tape (TAPE15).                                     C
C                                                                               C
C     Subroutine PRNTSM is called by: MODIF and CONDENSE                        C
C                                                                               C
C     Subroutine PRNTSM makes no external calls.                                C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
```

```
      DIMENSION AMAG(NN,NN),CSR(NN,NN),CSI(NN,NN)
      COMPLEX CSM(NN,NN)
      WRITE(15,5)
      WRITE(16,6)
    5 FORMAT(//,'REAL PARTS:  ',/)
    6 FORMAT(//,'IMAGINARY PARTS:  ',/)
      DO 10 I=1,NN
        DO 10 J=1,NN
          CSR(I,J)=REAL(CSM(I,J))
          CSI(I,J)=AIMAG(CSM(I,J))
          AMAG(I,J)=SQRT(REAL(CSM(I,J))**2.+AIMAG(CSM(I,J))**2.)
          IF(ABS(CSR(I,J)).LT.G*0.0001) CSR(I,J)=0.
          IF(ABS(CSI(I,J)).LT.G*0.00001) CSI(I,J)=0.
   10 CONTINUE
      WRITE(15,*)
      WRITE(15,*)
      IF(ICODE.EQ.4) WRITE(15,*)'AFTER ISO16:'
      IF(ICODE.EQ.5) WRITE(15,*)'AFTER MODIF:'
      IF(ICODE.EQ.6) WRITE(15,*)'AFTER CONDENSE:'
      WRITE(15,14)NN,NN,IEL,IM-1
      WRITE(16,*)
      WRITE(16,*)
      IF(ICODE.EQ.4) WRITE(16,*)'AFTER ISO16:'
      IF(ICODE.EQ.5) WRITE(16,*)'AFTER MODIF:'
      IF(ICODE.EQ.6) WRITE(16,*)'AFTER CONDENSE:'
      WRITE(16,14)NN,NN,IEL,IM-1
   14 FORMAT(1X,'STIFFNESS MATRIX (',I2,' x ',I2,') FOR ELEMENT',I3,
     2       ' AT WAVENUMBER ',I2,':')
      WRITE(15,15)
      WRITE(16,15)
   15 FORMAT(/,1X,'STIFFNESS MATRIX COLS. 1-8:',/)
      DO 25 I=1,NN
      WRITE(15,20) (CSR(I,J),J=1,8)
   25 WRITE(16,20) (CSI(I,J),J=1,8)
   20 FORMAT(1X,8(E15.6,2X))
      WRITE(15,22)
      WRITE(16,22)
   22 FORMAT(/,1X,'STIFFNESS MATRIX, COLS. 9-16:',/)
      DO 21 I=1,NN
      WRITE(15,20) (CSR(I,J),J=9,16)
   21 WRITE(16,20) (CSI(I,J),J=9,16)
      WRITE(15,31)
      WRITE(16,31)
   31 FORMAT(/,1X,'STIFFNESS MATRIX, COLS. 17-24:',/)
      DO 23 I=1,NN
      WRITE(15,20) (CSR(I,J),J=17,24)
   23 WRITE(16,20) (CSI(I,J),J=17,24)
      IF(NN.EQ.24) GOTO 99
      WRITE(15,32)
      WRITE(16,32)
   32 FORMAT(/,1X,'STIFFNESS MATRIX, COLS. 25-32:',/)
      DO 24 I=1,NN
      WRITE(15,20) (CSR(I,J),J=25,32)
   24 WRITE(16,20) (CSI(I,J),J=25,32)
      WRITE(15,33)
      WRITE(16,33)
   33 FORMAT(/,1X,'STIFFNESS MATRIX, COLS. 33-40:',/)
      DO 26 I=1,NN
      WRITE(15,20) (CSR(I,J),J=33,40)
   26 WRITE(16,20) (CSI(I,J),J=33,40)
      WRITE(15,34)
      WRITE(16,34)
   34 FORMAT(/,1X,'STIFFNESS MATRIX, COLS. 41-48:',/)
      DO 27 I=1,NN
      WRITE(15,20) (CSR(I,J),J=41,48)
   27 WRITE(16,20) (CSI(I,J),J=41,48)
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SYMSM(A,C,NN,IEL,ELG,IC1,IC2)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine SYMSM is used to check the symmetry of stiffne s       C
C     matrices at any desired step in the execution.                    C
C                                                                       C
C     IC1  =  0:  Real matrix                                           C
C             1:  Complex matrix                                        C
C                                                                       C
C     IC2  =  4:  After ISO16                                           C
C             5:  After MODIF                                           C
C             6:  After CONDENSE                                        C
C                                                                       C
C     Subroutine SYMSM is called by: MODIF, & CONDENSE                  C
C                                                                       C
C     Subroutine SYMSM makes no external calls.                         C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION A(NN,NN)
      COMPLEX C(NN,NN)
      COMMON /TOL/ TOLSM
C
C     Real Matrices
C
      IF(IC1.EQ.0) THEN
        DO 100 I=1,NN
          DO 100 J=1,NN
            IF(I.EQ.J) GOTO 100
            ADIFF=ABS(A(I,J)-A(J,I))
            IF(ADIFF/ABS(A(I,J)).GT.TOLSM.AND.A(I,J).GT.TOLSM*ELG)THEN
              WRITE(6,*)
              WRITE(6,*)
              IF(IC2.EQ.4) WRITE(6,*)'AFTER ISO16:'
              IF(IC2.EQ.5) WRITE(6,*)'AFTER MODIF:'
              IF(IC2.EQ.6) WRITE(6,*)'AFTER CONDENSE:'
              WRITE(6,101) IEL,I,J,A(I,J),J,I,A(J,I),ADIFF
            ELSE
            ENDIF
  100   CONTINUE
      ELSEIF(IC1.EQ.1) THEN
C
C     Complex matrices
C
        DO 200 I=1,NN
          DO 200 J=1,NN
            IF(I.EQ.J) GOTO 200
C
            AMAG1=SQRT(REAL(C(I,J))**2.+AIMAG(C(I,J))**2.)
            AMAG2=SQRT(REAL(C(J,I))**2.+AIMAG(C(J,I))**2.)
            DIFF1=ABS(REAL(C(I,J))-REAL(C(J,I)))
            DIFF2=ABS(AIMAG(C(I,J))-AIMAG(C(J,I)))
            DIFF3=ABS(AMAG1-AMAG2)
C
C           Disregard components with low magnitudes
C
            IF(AMAG1.LE.TOLSM*ELG) GOTO 200
C
C           Check real parts
C
            IF(REAL(C(I,J)).LE.ELG*TOLSM) GOTO 198
            IF(DIFF1/ABS(REAL(C(I,J))).GT.TOLSM)THEN
              WRITE(6,*)
              WRITE(6,*)
              IF(IC2.EQ.4) WRITE(6,*)'AFTER ISO16:'
              IF(IC2.EQ.5) WRITE(6,*)'AFTER MODIF:'
              IF(IC2.EQ.6) WRITE(6,*)'AFTER CONDENSE:'
              WRITE(6,101)IEL,I,J,REAL(C(I,J)),J,I,
     2                       REAL(C(J,I)),DIFF1
              STOP
```

```
            ENDIF
  198       CONTINUE
C
C             Check imaginary parts
C
            IF(ABS(AIMAG(C(I,J))).LE.ELG*TOLSM) GOTO 199
            IF(DIFF2/ABS(AIMAG(C(I,J))).GT.TOLSM)THEN
              WRITE(6,*)
              WRITE(6,*)
              IF(IC2.EQ.4) WRITE(6,*)'AFTER ISO16:'
              IF(IC2.EQ.5) WRITE(6,*)'AFTER MODIF:'
              IF(IC2.EQ.6) WRITE(6,*)'AFTER CONDENSE:'
              WRITE(6,102)IEL,I,J,AIMAG(C(I,J)),J,I,
     2                    AIMAG(C(J,I)),DIFF2
              STOP
            ENDIF
  199       CONTINUE
C
C             Check magnitudes
C
            IF(DIFF3/AMAG1.GT.TOLSM)THEN
              WRITE(6,*)
              WRITE(6,*)
              IF(IC2.EQ.4) WRITE(6,*)'AFTER ISO16:'
              IF(IC2.EQ.5) WRITE(6,*)'AFTER MODIF:'
              IF(IC2.EQ.6) WRITE(6,*)'AFTER CONDENSE:'
              WRITE(6,103)IEL,I,J,AMAG1,J,I,AMAG2,DIFF3
              STOP
            ENDIF
  200   CONTINUE
      ELSE
        WRITE(6,*)'PROBLEM IN SYMSM WITH IC1'
        STOP
      ENDIF
  101 FORMAT(//,1X,'**UNSYMMETRIC REAL PART OF 3-D ',
     1            'STIFFNESS MATRIX:',//,
     2           'ELEMENT: ',I5,10X,
     3       'SM(',I2,',',I2,')=',E15.8,'  SM(',I2,',',I2,')=',E15.8,/,
     4           'DIFFERENCE=',E10.3,//)
  102 FORMAT(//,1X,'**UNSYMMETRIC IMMAGINARY PART OF 3-D ',
     1            'STIFFNESS MATRIX:',//,
     2           'ELEMENT: ',I5,10X,
     3       'SM(',I2,',',I2,')=',E15.8,'i SM(',I2,',',I2,')=',E15.8,
     4       ' i',/,'DIFFERENCE=',E10.3,//)
  103 FORMAT(//,1X,'**UNSYMMETRIC MAGNITUDES OF 3-D ',
     1            'STIFFNESS MATRIX:',//,
     2           'ELEMENT: ',I5,10X,
     3       'SM(',I2,',',I2,')=',E15.8,'  SM(',I2,',',I2,')=',E15.8,/,
     4           'DIFFERENCE=',E10.3,//)
      RETURN
      END
C
C
C
C******************** FRONTAL SOLVER SUBROUTINES *********************
C
C
C
      SUBROUTINE SOLVE(A,IM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                    C
C     Subroutine SOLVE is a frontal solver originally written by     C
C     Prof. Johnson at the University of Texas at Austin.  Since     C
C     that time, the solver was modified by several graduate         C
C     students, some under the direction of Prof. Becker in EM.      C
C                                                                    C
C     Subroutine SOLVE begins the frontal solution technique by      C
C     calling one of the independent solvers for a full or upper     C
C     triangle stiffness matrix.                                     C
```

```
C                                                                       C
C     Subroutine SOLVE is called by: MAIN                               C
C                                                                       C
C     Subroutine SOLVE calls: COMPLT for a symmetric matrix             C
C                                   (upper triangle)                    C
C                             RESOL for an non-symmetric matrix         C
C                                   (full matrix)                       C
C     Variable list:                                                    C
C         IAB       Sub-index for (A(IND(13)))                        C
C         IAE       Sub-index for (A(IND(13)))                        C
C         IAF       Sub-index for (A(IND(13)))                        C
C         IAL       Sub-index for (A(IND(13)))                        C
C         IAM       Sub-index for (A(IND(13)))                        C
C         IAN       Sub-index for (A(IND(13)))                        C
C         IB                                                          C
C         IFB       Number of writes to NTAPEB                        C
C         IFL       Number of writes to NTAPEL                        C
C         IFU       Number of writes to NTAPEU                        C
C         IL                                                          C
C         IU                                                          C
C         IERR      Error code                                        C
C         IFG       Solution status code                              C
C         IPRINT    Print code                                        C
C         IRESOL    Solution code                                     C
C         ISYM      Symmetry code                                     C
C         IWRT                                                        C
C         MA        Total memory allocated for solution               C
C         MB        Memory length for RHS buffer                      C
C         MDOF      Maximum number of DOF's per element (=NDOFE2)     C
C         MBUF      Memory length for LHS buffer                      C
C         MELEM     Memory length for Element                         C
C         MFW                                                         C
C         MFWR      Memory length for Front                           C
C         MKF                                                         C
C         MLDEST    Memory length for destination vector              C
C         MW                                                          C
C         N         Integer array                                     C
C         NNEGP     Number of negative pivot points (real part)       C
C         NPOSP     Number of positive pivot points (real part)       C
C         NRHSF                                                       C
C         NUMEL2    Number of elements (condensed)                    C
C         NRHS      Number of RHS's                                   C
C         NTAPEB    Tape number for B                                 C
C         NTAPEL    Tape number for L (loads)                         C
C         NTAPEU    Tape number for U (displacements)                 C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON/CNTL/ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,MA,
     .            IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .            IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .            MELEM,MFWR,MB,MDOF,MFW,MLDEST
      COMPLEX A(1)
      NNEGP = 0
      NPOSP = 0
      CALL COMPLT(A,IM)
      RETURN
      END
C
C
```

```
C *** SYMMETRIC MATRIX ROUTINES (LOWER TRIANGLE)
C
      SUBROUTINE COMPLT(A,IM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine COMPLT initiates the solution technique for a        C
C     symmetric stiffness matrix.  Only the top half of the           C
C     stiffness matrix is read.  Forward elimination of the LHS       C
C     and RHS are performed followed by backsubstitution.             C
C                                                                     C
C     Fixed parameters:                                               C
C                       ISYM  =  1                                    C
C                       IRESOL = 0                                    C
C                       NRHS =   1                                    C
C                                                                     C
C     Subroutine COMPLT is called by: SOLVE                           C
C                                                                     C
C     Subroutine COMPLT calls: SECOND, FRWCP, and BCKWRD              C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON/CNTL/ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     .          MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .          IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .          MELEM,MFWR,MB,MDOF,MFW,MLDEST
      COMPLEX A(1)
      CALL SECOND(T0)
      IERR = 1
      N = NUMEL2+MLDEST+2*MDOF
      IF(ISYM .GT. 1) GO TO 10
      MELEM = (MDOF*(MDOF+1))/2+MDOF*NRHS
      MKF = (MFW*(MFW+1))/2
      GO TO 20
   10 MELEM = MDOF*(MDOF+NRHS)
      MKF = MFW*MFW
   20 MFWR = MKF+MFW*NRHS
      MW = MELEM+MFWR
      MBUF = MA-MW-N
      IF(ISYM .GT. 1) GO TO 40
      IF(IPRINT .NE. 0 .AND. IM. EQ. 1) PRINT 1000
      GO TO 50
   40 IF(IPRINT .NE. 0 .AND. IM. EQ. 1) PRINT 1010
      IF(ISYM.EQ.2.AND.IPRINT.NE.0.AND.IM.EQ.1) PRINT 1020
   50 IF(IPRINT .NE. 0.AND.IM.EQ.1) PRINT 1030,N,MELEM,MFWR,MBUF,MA
      IF(MBUF .LT. MFW+NRHS) GO TO 70
      IAL = 1+NUMEL2
      IAM = IAL+MLDEST
      IAN = IAM+MDOF
      IAE = IAN+MDOF
      IAF = IAE+MELEM
      IAB = IAF+MFWR
      CALL FRWCP(A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IAB))
      CALL SECOND(TF)
      DT = TF-T0
      IF(IPRINT .NE. 0. AND. IM. EQ. 1) PRINT 1040,DT
      IF(IPRINT.NE.0.AND.IFU.NE.0.AND.IM.EQ.1) PRINT 1043,IFU
      IF(IPRINT.NE.0.AND.IFL.NE.0.AND.IM.EQ.1) PRINT 1045,IFL
      IF(IERR .NE. 1) RETURN
      IF(NRHS .EQ. 0) GO TO 60
      CALL BCKWRD(A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IAB),
     .            A(IAB))
      CALL SECOND(TB)
      DT = TB-TF
   60 RETURN
   70 IERR = 6
      PRINT 1060
      STOP
      RETURN
C
```

```
 1000 FORMAT(/,1X,'FRONTAL SOLVER INFORMATION:',/,
     +        5X,29HSYMMETRIC FORWARD ELIMINATION     ,/)
 1010 FORMAT(/,1X,'FRONTAL SOLVER INFORMATION:',/,
     +        5X,32HUNSYMMETRIC FORWARD ELIMINATION  ,/)
 1020 FORMAT(       5X,22HRESOLUTION INACTIVATED          ,/)
 1030 FORMAT(     4X,21H       INTEGER ARRAY:            ,I7,/,
     .            4X,21H          REAL ARRAY             ,/,
     .            4X,21H             ELEMENT:            ,I7,/,
     .            4X,21H               FRONT:            ,I7,/,
     .            4X,21H              BUFFER:            ,I7,/,
     .            4X,21H       TOTAL STORAGE:            ,I7)
 1040 FORMAT(     10X,29HTIME IN FORWARD ELIMINATION:     ,F9.3,/)
 1043 FORMAT(     10X,18HWRITES TO NTAPEU:               ,I4,/)
 1045 FORMAT(     10X,18HWRITES TO NTAPEL:               ,I4,/)
 1060 FORMAT(2(/), 5X,32HERROR: NOT ENOUGH ROOM IN BUFFER,/,
     +           5X,'PROGRAM TERMINATED',/)
      END
C
C
C
      SUBROUTINE FRWCP(LELM,LDEST,MDEST,NDEST,ELEM,FRNT,BUF)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                   C
C     Subroutine FRWCP performs the forward elimination of both the C
C     LHS and RHS.                                                  C
C                                                                   C
C     Subroutine FRWCP is called by: COMPLT                         C
C                                                                   C
C     Subroutine FRWCP calls: SOLIN, DEST, SYMASM, UNSASM, SEMRHS,  C
C                             TOUT, UNSELM, SYMELM, and ELMRHS      C
C                                                                   C
    C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON/CNTL/ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,MA,
     .            IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .            IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .            MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION LDEST(1),MDEST(1),NDEST(1),LELM(1)
      COMPLEX ELEM(1),FRNT(1),BUF(1)
      REWIND NTAPEU
      IF(ISYM .EQ. 3) REWIND NTAPEL
      IFU = 0
      IFL = 0
      NRHSF = NRHS
      IU = 1
      IL = MBUF
      NFW = 0
      LFW = 0
      DO 200 IEL=1,NUMEL2
        CALL SOLIN(IEL,3,NRHS,NUMDES,LDEST,ELEM)
        CALL DEST(NUMDES,LDEST,NFW,NDOF,NE,MDEST,NDEST)
        IF(LFW .GT. NFW) NFW = LFW
        IF(ISYM .EQ. 1) CALL SYMASM(NDOF,LFW,NFW,MDEST,ELEM,FRNT)
        IF(ISYM .GT. 1) CALL UNSASM(NDOF,LFW,NFW,MDEST,ELEM,FRNT)
        KFW = NFW
        IF(NRHS .EQ. 0) GO TO 30
        IF(ISYM .GT. 1) GO TO 10
        MKE = (NDOF*(NDOF+1))/2
        GO TO 20
   10   MKE = NDOF*NDOF
   20   CALL SEMRHS(LFW,NFW,NDOF,NRHS,MFW,MDEST,ELEM(MKE+1),FRNT(MKF+1))
   30   IF(NE .EQ. 0) GO TO 155
        DO 150 IE=1,NE
          N = IU+NFW+NRHS-1
          IF(N .LE. IL) GO TO 40
          CALL TOUT(1,IU,IFU,NTAPEU,BUF)
          IU = 1
   40     M = IU
          IF(ISYM .EQ. 3) GO TO 50
          IF(ISYM .EQ. 2) CALL UNSELM(IEL,KFW,NFW,NDEST(IE),FRNT,
```

```
     *                                    BUF(IU))
           IF(ISYM .EQ. 1) CALL SYMELM(IEL,NFW,NDEST(IE),FRNT,BUF(IU))
           IU = IU+NRHS+NFW
           GO TO 70
    50     N = IU+NFW+NRHS-1
           IF(N .LE. IL) GO TO 60
           CALL TOUT(IL,MBUF,IFL,NTAPEL,BUF(IL+1))
           IL = MBUF
    60     CALL UNSELM(IEL,KFW,NFW,NDEST(IE),FRNT,BUF(IU))
           IU = IU+NRHS+NFW
    70     IF(IERR .EQ. 1) GO TO 75
           PRINT 1000,IEL
           RETURN
    75     IF(NRHS .EQ. 0) GO TO 90
           IF(ISYM .GT.1) GO TO 80
           CALL ELMRHS(NFW,MFW,NRHS,NDEST(IE),1,FRNT(MKF+1),BUF(M),
          .                BUF(M+NFW))
           GO TO 120
    80     CALL ELMRHS(NFW,MFW,NRHS,NDEST(IE),KFW,FRNT(MKF+1),FRNT(NFW),
          .                BUF(M+NFW))
           IF(ISYM .EQ. 2) GO TO 120
    90     IF(ISYM .NE. 3) GO TO 120
           IF(IL-NFW+1 .GE. N) GO TO 100
           CALL TOUT(1,IU,IFU,NTAPEU,BUF)
           IU = 1
   100     M = NFW
           N = NFW-1
           DO 110 J=1,N
             BUF(IL) = FRNT(M)
             IL = IL-1
   110     M = M+KFW
   120     CONTINUE
   150   NFW = NFW-1
   155   CONTINUE
         LFW = NFW
         LELM(IEL) = LFW
         IF(ISYM .EQ. 1  .OR. NE .EQ. 0) GO TO 200
         N = KFW
         M = NFW+1
         DO 170 I=2,NFW
           DO 160 J=1,NFW
             FRNT(M) = FRNT(N+J)
   160     M = M+1
   170   N = N+KFW
   200 CONTINUE
       IB = IU
       IF(IWRT .EQ. 0  .AND.  IFU .EQ. 0) GO TO 210
       CALL TOUT(1,IU,IFU,NTAPEU,BUF)
       BACKSPACE NTAPEU
   210 IF(ISYM .NE. 3) RETURN
       IF(IWRT .EQ. 0  .AND.  IFL .EQ. 0) RETURN
       CALL TOUT(IL,MBUF,IFL,NTAPEL,BUF(IL+1))
       RETURN
  1000 FORMAT(2(/), 5X,42HERROR: ZERO PIVOT IN ELEMENT:
      .                   ,I5)
       END
C
C
C
       SUBROUTINE SYMASM(NDOF,LFWX,NFWX,MDEST,ELLHS,FLHS)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C      Subroutine SYMASM assembles the LHS for symmetric matrices.              C
C                                                                               C
C      Subroutine SYMASM is called by: FRWCP                                    C
C                                                                               C
C      Subroutine SYMASM makes no external calls.                               C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
```

```
      DIMENSION MDEST(1)
      COMPLEX ELLHS(1),FLHS(1)
      LFW = LFWX
      NFW = NFWX
      IF(NFW .EQ. LFW) GO TO 20
      MI = (LFW*(LFW+1))/2+1
      MJ = (NFW*(NFW+1))/2
      DO 10 I=MI,MJ
   10 FLHS(I) = (0.,0.)
   20 N = 1
      DO 50 I=1,NDOF
        MI = MDEST(I)
        DO 50 J=1,I
          MJ = MDEST(J)
          MK = MAX0(MI,MJ)
          MJ = MIN0(MI,MJ)
          MK = (MK*(MK-1))/2+MJ
          FLHS(MK) = FLHS(MK)+ELLHS(N)
   50 N = N+1
      RETURN
      END
C
C
C
      SUBROUTINE SYMELM(IEL,NFWX,IDX,FLHS,U)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine SYMELM eliminates one equation (ID) for symmetric              C
C     matrices.                                                                 C
C                                                                               C
C     Subroutine SYMELM is called by: FRWCP                                     C
C                                                                               C
C     Subroutine SYMELM makes no external calls.                                C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ IDUM(9),IPRINT,IERR,NNEGP,NPOSP,IIDUM(16)
      COMPLEX FLHS(1),U(1),PIVOTC,S
      ID = IDX
      NFW = NFWX
      MP=(ID*(ID+1))/2
      IDM = ID-1
      IDP = ID+1
      M = MP-ID+1
      K = 1
C
C     OK to use real part for comparisons???
C
      PIVOT = REAL(FLHS(MP))
      PIVOTC = FLHS(MP)
      IF(IPRINT .GE. 2) PRINT 200,IEL,NFW,ID,PIVOTC
  200 FORMAT(5X,17HIEL,NFW,ID,PIVOTC  ,3I5,E13.4,'+',E13.4,'I')
      U(ID) = PIVOTC
      IF(ABS(PIVOT) .LE. 1.E-30) GO TO 90
      IF(PIVOT .LT. 0.) NNEGP = NNEGP+1
      IF(PIVOT .GT. 0.) NPOSP = NPOSP+1
      IF(IDM .EQ. 0) GO TO 30
      DO 20 I=1,IDM
        S = FLHS(M)
        U(I) = S/PIVOTC
        DO 10 J=1,I
          FLHS(K) = FLHS(K)-S*U(J)
   10   K=K+1
   20 M=M+1
   30 M=MP
      K = 0
      IF(IDP .GT. NFW) GO TO 100
      DO 60 I=IDP,NFW
        NN = M-ID
        M = M+ID+K
```

```
      N = M-ID
      S = FLHS(M)
      U(I) = S/PIVOTC
      IF(IDM .EQ. 0) GO TO 50
      DO 40 J=1,IDM
   40 FLHS(NN+J) = FLHS(N+J)-S*U(J)
   50 NN = NN-1
      DO 55 J=IDP,I
   55 FLHS(NN+J) = FLHS(N+J)-S*U(J)
   60 K=K+1
      GO TO 100
   90 IERR = 2
  100 RETURN
      END
C
C
C
      SUBROUTINE UNSASM(NDOF,LFWX,NFWX,MDEST,ELLHS,FLHS)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                       C
C     Subroutine UNSASM assembles the LHS for unsymmetric matrices.     C
C                                                                       C
C     Subroutine UNSASM is called by: FRWCP                             C
C                                                                       C
C     Subroutine UNSASM makes no external calls.                        C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION MDEST(1)
      COMPLEX ELLHS(1),FLHS(1)
      LFW = LFWX
      NFW = NFWX
      IF(NFW .EQ. LFW) GO TO 40
      MI = LFW*NFW+1
      MJ = NFW*NFW
      MK = LFW*LFW+1
      DO 10 I=MI,MJ
   10 FLHS(I) = 0.
      IF(LFW .EQ. 0) GO TO 40
      MJ = NFW-LFW
      DO 30 I=1,LFW
        DO 20 J=1,MJ
          MI = MI-1
   20   FLHS(MI) = (0.,0.)
        DO 30 J=1,LFW
          MI = MI-1
          MK = MK-1
   30 FLHS(MI) = FLHS(MK)
      MI = NFW*NFW
   40 N = 1
      DO 50 I=1,NDOF
        MI = MDEST(I)
        MK = (MI-1)*NFW
        DO 50 J=1,NDOF
          MJ = MDEST(J)
          ML = MK+MJ
          FLHS(ML) = FLHS(ML)+ELLHS(N)
   50 N = N+1
      RETURN
      END
C
C
C
```

```
      SUBROUTINE UNSELM(IEL,KFWX,NFWX,IDX,FLHS,U)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine UNSELM eliminates one equation (ID) for unsymmetric  C
C     matrices.                                                       C
C                                                                     C
C     Subroutine UNSELM is called by: FRWCP                           C
C                                                                     C
C     Subroutine UNSELM makes no external calls.                      C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ IDUM(9),IPRINT,IERR,NNEGP,NPOSP,IIDUM(16)
      COMPLEX FLHS(1),U(1),PIVOTC,S
      ID = IDX
      KFW = KFWX
      NFW = NFWX
      IDM = ID-1
      IDP = ID+1
      K = IDM*KFW
      MP = K+ID
      PIVOTC = FLHS(MP)
      PIVOT = REAL(FLHS(MP))
      IF(IPRINT .GE. 2) PRINT 200, IEL,NFW,ID,PIVOTC
  200 FORMAT(5X,17HIEL,NFW,ID,PIVOTC  ,3I5,E13.4,'+',E13.4,'i')
      IF(ABS(PIVOT) .LE. 1.E-30) GO TO 90
      IF(PIVOT .LT. 0.) NNEGP = NNEGP+1
      IF(PIVOT .GT. 0.) NPOSP = NPOSP+1
      DO 5 I=1,NFW
    5 U(I) = FLHS(K+I)
      K = 0
      IF(IDM .EQ. 0) GO TO 40
      DO 30 I=1,IDM
        S = FLHS(ID+K)/PIVOTC
        DO 10 J=1,IDM
          M = J+K
   10   FLHS(M) = FLHS(M)-S*U(J)
        M = K-1
        IF(IDP .GT. NFW) GO TO 25
        DO 20 J=IDP,NFW
   20   FLHS(J+M) = FLHS(J+K)-S*U(J)
   25   K = K+KFW
   30 FLHS(K-KFW+NFW) = S
   40 K=K+KFW
      IF(IDP .GT. NFW) GO TO 100
      DO 70 I=IDP,NFW
        S = FLHS(ID+K)/PIVOTC
        M = K-KFW
        IF(IDM .EQ. 0) GO TO 55
        DO 50 J=1,IDM
   50   FLHS(J+M) = FLHS(K+J)-S*U(J)
   55   M=M-1
        DO 60 J=IDP,NFW
   60   FLHS(M+J) = FLHS(K+J)-S*U(J)
        FLHS(K-KFW+NFW) = S
   70 K = K+KFW
      GO TO 100
   90 IERR = 2
  100 CONTINUE
      RETURN
      END
C
C
C *** ROUTINES FOR NON-SYMMETRIC MATRICES (FULL MATRIX)
C
C
C *** (REMOVED SINCE NOT USED) ***
C
C
```

```
C *** ROUTINES COMMON TO BOTH SOLUTION METHODS
C
C
      SUBROUTINE BCKWRD(LELM,LDEST,MDEST,NDEST,ELEM,FRNT,B,U)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine BCKWRD is used to perform the backsubstitution.                C
C                                                                               C
C     Subroutine BCKWRD is called by:  COMPLT, RESOL                            C
C                                                                               C
C     Subroutine BCKWRD calls: SOLIN, DEST, ELMSOL, SOLOUT                      C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ ISYM,NUMEL2,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,MA,
     .              IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     .              IB,IUU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     .              MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION LDEST(1),MDEST(1),NDEST(1),LELM(1)
      COMPLEX ELEM(1),FRNT(1),B(1),U(1)
      IU = IUU
      JEL = NUMEL2+1
      IB = IB-NRHS
      DO 100 IEL=1,NUMEL2
        JEL = JEL-1
        CALL SOLIN(JEL,1,NRHS,NUMDES,LDEST,ELEM)
        CALL DEST(NUMDES,LDEST,NFW,NDOF,NE,MDEST,NDEST)
        IF(JEL .EQ. 1) GO TO 7
        LFW = LELM(JEL-1)
        IF(LFW .GT. NFW) NFW = LFW
    7   CONTINUE
        NFW = NFW-NE+1
        IF(NE .EQ. 0) GO TO 35
        J = NE+1
        DO 30 I=1,NE
          J = J-1
          IF(IU .GT. 1) GO TO 10
          BACKSPACE NTAPEU
          READ(NTAPEU) IU,(U(II),II=1,IU)
          BACKSPACE NTAPEU
          IU = IU+1
   10     IU = IU-NFW-NRHSF
          IF(IRESOL .EQ. 1) GO TO 20
          N = IU+NFW
          CALL ELMSOL(NFW,MFW,NRHS,NDEST(J),U(IU),U(N),FRNT(1))
          GO TO 30
   20     IF(IB .GE. 1) GO TO 25
          BACKSPACE NTAPEB
          READ(NTAPEB) IB,(B(II),II=1,IB)
          BACKSPACE NTAPEB
          IB = IB-NRHS+1
   25     CONTINUE
          CALL ELMSOL(NFW,MFW,NRHS,NDEST(J),U(IU),B(IB),FRNT(1))
          IB = IB-NRHS
   30   NFW = NFW+1
   35   DO 40 I=1,NDOF
          K = 0
          L = 0
          M = MDEST(I)
          DO 40 J=1,NRHS
            ELEM(K+I) = FRNT(L+M)
            K = K+NDOF
   40   L = L+MFW
        CALL SOLOUT(JEL,NDOF,NRHS,ELEM)
  100 CONTINUE
      RETURN
      END
C
C
C
```

```
      SUBROUTINE DEST(ND,LDEST,NFW,NDOF,NEE,MDEST,NDEST)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine DEST converts DEST vectors to DOF DEST. vectors.     C
C     The equations to be eliminated are written to NDEST giving      C
C     the current location in the front.                              C
C                                                                     C
C     Subroutine DEST is called by: FRWCP, FRWRS, & BKWRD             C
C                                                                     C
C     Subroutine DEST makes no external calls.                        C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ IDUM(9),IPRINT,IIDUM(19)
      DIMENSION LDEST(1),MDEST(1),NDEST(1)
C
C     The following statement appears to be a function.  It therefore
C     has been replaced with the equation where called.
C
      MODR(I,J) = I-I/J*J
      NFW = 0
      KM = 1
      KN = 1
      NDOF = 0
      NE = 0
      DO 50 I=1,ND
        M = MODR(LDEST(I),10)
        M2 = LDEST(I)-LDEST(I)/10*10
        N = MODR(LDEST(I),100)/10
        N2 = (LDEST(I)-LDEST(I)/100*100)/10
        IF(ABS(M-M2).GT.0.5.OR.ABS(N-N2).GT.0.5) THEN
          WRITE(6,*)'PROBLEM IN DEST WITH MODR'
          STOP
        ELSE
        ENDIF
        NDOF = NDOF+N
        IF(M .GE. 1) NE = NE+N
        L = LDEST(I)/100-1
        DO 10 J=1,N
          MDEST(KM) = L+J
          IF(M .EQ. 0) GO TO 10
          NDEST(KN) = L+J
          KN = KN+1
   10   KM = KM+1
        L = MDEST(KM-1)
        IF(L .GT. NFW) NFW = L
   50 CONTINUE
      IF(NE .EQ. 0) GO TO 80
      DO 70 I=1,NE
        J = I+1
        DO 70 L=J,NE
          IF(NDEST(I) .LT. NDEST(L)) NDEST(L) = NDEST(L)-1
   70 CONTINUE
   80 NEE = NE
      IF(IPRINT .LE. 2) RETURN
      PRINT 1000,(LDEST(I),I=1,ND)
      PRINT 1010,(MDEST(I),I=1,NDOF)
      PRINT 1020,(NDEST(I),I=1,NE)
 1000 FORMAT(/X*IN DEST: NODAL DESTINATION VECTORS*,10I7,10(/,35X,10I7))
 1010 FORMAT(11X,*DOF DESTINATION VECTORS*10I7,10(/,35X,10I7))
 1020 FORMAT(9X*ELIM. DESTINATION VECTORS*10I7,10(/,35X,10I7))
      RETURN
      END
C
C
C
```

```
      SUBROUTINE ELMRHS(NFW,MFW,NRHS,ID,INC,FRHS,U,B)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine ELMRHS eliminates RHS's for equation (ID).           C
C                                                                     C
C     Subroutine ELMRHS is called by: FRWCP & FRWRS                   C
C                                                                     C
C     Subroutine ELMRHS makes no external calls.                      C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMPLEX FRHS(1),U(1),B(1),S
      IDM = ID-1
      IDP = ID+1
      IM = 0
      DO 50 IN = 1,NRHS
        IU = 1
        S = FRHS(IM+ID)
        B(IN) = S
        IF(IDM .EQ. 0) GO TO 25
        DO 20 I=1,IDM
          II = IM+I
          FRHS(II) = FRHS(II)-S*U(IU)
   20   IU=IU+INC
   25   IF(ISYM .EQ. 1) IU = IU+1
        IF(IDP .GT. NFW) GO TO 50
        DO 30 I=IDP,NFW
          II = IM+I
          FRHS(II-1) = FRHS(II)-S*U(IU)
   30   IU = IU+INC
   50 IM = IM+MFW
      RETURN
      END
C
C
C
      SUBROUTINE ELMSOL(NFW,MFW,NRHS,IDX,U,B,X)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine ELMSOL calculates the solution for one DOF as        C
C     specified by (ID).                                              C
C                                                                     C
C     Subroutine ELMSOL is called by: BKWRD                           C
C                                                                     C
C     Subroutine ELMSOL makes no external calls.                      C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMPLEX U(1),B(1),X(1),F1,F2,S
      ID = IDX
      IDM = ID-1
      IDP = ID+1
      IF(ISYM .GT. 1) GO TO 5
      F1 = U(ID)
      F2 = (1.,0.)
      GO TO 7
    5 F1 = (1.,0.)
      F2 = U(ID)
    7 CONTINUE
      DO 40 IN=1,NRHS
        IU = NFW
        JA = (IN-1)*MFW
        IA = JA+NFW-1
        S = B(IN)/F1
        IF(IDP .GT. NFW) GO TO 20
        DO 10 I=IDP,NFW
          X(IA+1) = X(IA)
          S = S-U(IU)*X(IA)
          IA = IA-1
```

```
   10   IU = IU-1
   20   IU = IU-1
        IF(IDM .LT. 1) GO TO 40
        DO 30 I=1,IDM
          S = S-U(IU)*X(IA)
          IA = IA-1
   30   IU = IU-1
   40 X(JA+ID) = S/F2
      RETURN
      END
C
C
C
      SUBROUTINE SEMRHS(LFW,NFW,NDOF,NRHS,MFW,MDEST,ELRHS,FRHS)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                      C
C     Subroutine SEMRHS assembles the RHS for a full symmetric         C
C     matrix.                                                          C
C                                                                      C
C     Subroutine SEMRHS is called by: FRWCP & FRWRS                    C
C                                                                      C
C     Subroutine SEMRHS makes no external calls.                       C
C                                                                      C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION MDEST(1)
      COMPLEX ELRHS(1),FRHS(1)
      N = 1
      DO 70 IN=1,NRHS
        IA = (IN-1)*MFW
        IF(NFW .EQ. LFW) GO TO 15
        M = LFW+1
        DO 13 I=M,NFW
   13   FRHS(IA+I) = (0.,0.)
   15   CONTINUE
        DO 50 I=1,NDOF
          J = IA+MDEST(I)
          FRHS(J) = FRHS(J)+ELRHS(N)
   50   N = N+1
   70 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SOLIN(IEL,IFG,NRHS,NUMDES,LDEST,ELEM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                      C
C     Subroutine SOLIN is used to read in necessary LHS and RHS        C
C     data from Tape10.  Data is returned through ELEM.                C
C     Lengths LDEST and NUMDES are also returned.                      C
C                                                                      C
C     Subroutine SOLIN is called by: BCKWRD, FRWCP, FRWRS              C
C                                                                      C
C     Subroutine SOLIN makes no external calls.                        C
C                                                                      C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION LDEST(1)
      COMPLEX ELEM(1)
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /FLAGS/ IND(22)
      COMMON /UNIT/ NTAPE,NTAPEO
      COMMON /WORK/ IDEST(1)
      NUMDES=NNE2
      J=IND(12)+NNE2*(IEL-1)-1
      DO 10 I=1,NUMDES
        J=J+1
        LDEST(I)=IDEST(J)
   10 CONTINUE
C
```

```
C     Sort by IFG (call index)-
C         IFG=1: Used for BCKWRD
C         IFG=2: Used for FRWRS
C         IFG=3: Used for FRWCP
C
      IF(IFG.EQ.1) RETURN
c     NDOF=24
       IF(IFG.EQ.2) THEN
        LENGTH=NDOFE2*NRHS
      ELSE
        LENGTH=((NDOFE2*(NDOFE2+1))/2)+NDOFE2*NRHS
      ENDIF
C
C     Read appropriate data from NTAPE (Tape10)
C
      READ(NTAPE) (ELEM(I),I=1,LENGTH)
      RETURN
      END
C
C
C
      SUBROUTINE SOLOUT(IEL,NDOF,NRHS,ELEM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine SOLOUT is used to write the displacement vector      C
C     to C(IND(14)).                                                  C
C                                                                     C
C     Subroutine SOLOUT is called by: BCKWRD                          C
C                                                                     C
C     Subroutine SOLOUT makes no external calls.                      C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION IA(200000),IAA(70000),AA(70000)
      COMPLEX A(200000),ELEM(1)
      EQUIVALENCE (A(1),IA(1)),(AA(1),IAA(1))
      COMMON /CONDS/ NNE2,NDOFN2,NNT2,NDOFE2,NDOFT2
      COMMON /FLAGS/ IND(22)
      COMMON /REALA/ AA
      COMMON /WORK/ A
      MNNE=16
      J=IND(7)+MNNE*(IEL-1)-1
      M=0
      DO 20 I=1,NNE2
C
C       Use front face of front wall to get proper indexing in A
C
        NODE=IAA(J+I)
        K=IND(14)+NDOFN2*(NODE-1)-1
        DO 10 L=1,NDOFN2
          A(K+L)=ELEM(M+L)
   10   CONTINUE
        M=M+NDOFN2
   20 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE TIN(L,I,J,NT,B)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                     C
C     Subroutine TIN reads the RHS buffer tape.                       C
C                                                                     C
C     Subroutine TIN is called by: FRWRS                              C
C                                                                     C
C     Subroutine TIN makes no external calls.                         C
C                                                                     C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
```

```
      COMPLEX B(1)
      READ(NT) K,(B(II),II=1,K)
      IF(L .GT. 0) GO TO 5
      I = 1
      J = K
      RETURN
    5 I = K
      J = 1
      RETURN
      END
C
C
C
      SUBROUTINE TOUT(I,J,IF,NT,B)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine TOUT writes all buffers to tape.                               C
C                                                                               C
C     Subroutine TOUT is called by: FRWCP & FRWRS                               C
C                                                                               C
C     Subroutine TOUT makes no external calls.                                  C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      COMPLEX B(1)
      IF(J .EQ. I) RETURN
      K = J-I
      IF = IF+1
      WRITE(NT) K,(B(II),II=1,K)
      RETURN
      END
C
C
C
C********************* FFT SUBROUTINES *********************************
C
      SUBROUTINE FOUR2 (DATA,N,NDIM,ISIGN,IFORM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                               C
C     Subroutine FOUR2 is a FFT program originally written at MIT.             C
C     The arguments for FOUR2 are-                                              C
C                                                                               C
C               DATA:  array name                                               C
C               N:     array used in the subroutine with the                    C
C                      digitization rates (power of 2) for each                 C
C                      dimensions heading the array                             C
C               NDIM:  number of dimensions                                     C
C               ISIGN: SIGN of transform                                        C
C               IFORM-                                                          C
C                    =-1: convert from complex to real                          C
C                    = 0: convert from real to complex                          C
C                    = 1: stay same (real-real or complex-complex               C
C                                                                               C
C     Subroutine FOUR2 is called by:  MAIN                                      C
C                                                                               C
C     Subroutine FOUR2 calls: BITRV, COOL2, FIXRL                               C
C                                                                               C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION DATA(1), N(1)
      NTOT=1
      DO 10 IDIM=1,NDIM
 10   NTOT=NTOT*N(IDIM)
      IF (IFORM) 70,20,20
C
C     Convert from real to complex (iform >= 0)
C
 20   NREM=NTOT
      DO 60 IDIM=1,NDIM
        NREM=NREM/N(IDIM)
        NPREV=NTOT/(N(IDIM)*NREM)
```

```
      NCURR=N(IDIM)
      IF (IDIM-1+IFORM) 30,30,40
 30   NCURR=NCURR/2
 40   CALL BITRV (DATA,NPREV,NCURR,NREM)
      CALL COOL2 (DATA,NPREV,NCURR,NREM,ISIGN)
      IF (IDIM-1+IFORM) 50,50,60
 50   CALL FIXRL (DATA,N(1),NREM,ISIGN,IFORM)
      NTOT=(NTOT/N(1))*(N(1)/2+1)
 60   CONTINUE
      RETURN
C
C     Convert from complex to real (iform < 0)
C
 70   NTOT=(NTOT/N(1))*(N(1)/2+1)
      NREM=1
      DO 100 JDIM=1,NDIM
        IDIM=NDIM+1-JDIM
        NCURR=N(IDIM)
        IF (IDIM-1) 80,80,90
 80     NCURR=NCURR/2
        CALL FIXRL (DATA,N(1),NREM,ISIGN,IFORM)
        NTOT=NTOT/(N(1)/2+1)*N(1)
 90     NPREV=NTOT/(N(IDIM)*NREM)
        CALL BITRV (DATA,NPREV,NCURR,NREM)
        CALL COOL2 (DATA,NPREV,NCURR,NREM,ISIGN)
 100  NREM=NREM*N(IDIM)
      RETURN
      END
C
C
C
      SUBROUTINE BITRV (DATA,NPREV,N,NREM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                   C
C     Subroutine BITRV is part of the FFT program.  It shuffles     C
C     the data by 'bit reversal.'                                   C
C                                                                   C
C     DIMENSION DATA(NPREV,N,NREM)                                  C
C     DATA(I1,I2REV,I3) = DATA(I1,I2,I3), FOR ALL I1 FROM 1 TO NPREV, C
C     ALL I2 FROM 1 TO N (WHICH MUST BE A POWER OF TWO), AND ALL I3  C
C     FROM 1 TO NREM, WHERE I2REV-1 IS THE BITWISE REVERSAL OF I2-1. C
C     FOR EXAMPLE, N = 32, I2-1 = 10011 AND I2REV-1 = 11001.        C
C                                                                   C
C     Subroutine BITRV is called by:  FOUR2                         C
C                                                                   C
C     Subroutine BITRV makes no external calls.                     C
C                                                                   C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION DATA(1)
      IP0=2
      IP1=IP0*NPREV
      IP4=IP1*N
      IP5=IP4*NREM
      I4REV=1
      DO 60 I4=1,IP4,IP1
        IF (I4-I4REV) 10,30,30
 10     I1MAX=I4+IP1-IP0
        DO 20 I1=I4,I1MAX,IP0
          DO 20 I5=I1,IP5,IP4
            I5REV=I4REV+I5-I4
            TEMPR=DATA(I5)
            TEMPI=DATA(I5+1)
            DATA(I5)=DATA(I5REV)
            DATA(I5+1)=DATA(I5REV+1)
            DATA(I5REV)=TEMPR
 20     DATA(I5REV+1)=TEMPI
 30     IP2=IP4/2
 40     IF (I4REV-IP2) 60,60,50
 50     I4REV=I4REV-IP2
```

```
        IP2=IP2/2
        IF (IP2-IP1) 60,40,40
 60   I4REV=I4REV+IP2
      RETURN
      END
C
C
C
      SUBROUTINE COOL2 (DATA,NPREV,N,NREM,ISIGN)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                      C
C     Subroutine COOL2 is part of the FFT program.                     C
C                                                                      C
C                                                                      C
C     Subroutine COOL2 is called by:  FOUR2                            C
C                                                                      C
C     Subroutine COOL2 makes no external calls.                        C
C                                                                      C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION DATA(1)
      TWOPI=2.*(4.*ATAN(1.))*ISIGN
      IP0=2
      IP1=IP0*NPREV
      IP4=IP1*N
      IP5=IP4*NREM
      IP2=IP1
      NPART=N
 10   IF (NPART-2) 50,30,20
 20   NPART=NPART/4
      GO TO 10
C
C     DO A FOURIER TRANSFORM OF LENGTH TWO
C
 30   IP3=IP2*2
      DO 40 I1=1,IP1,IP0
        DO 40 I5=I1,IP5,IP3
          J0=I5
          J1=J0+IP2
          TEMPR=DATA(J1)
          TEMPI=DATA(J1+1)
          DATA(J1)=DATA(J0)-TEMPR
          DATA(J1+1)=DATA(J0+1)-TEMPI
          DATA(J0)=DATA(J0)+TEMPR
 40   DATA(J0+1)=DATA(J0+1)+TEMPI
      GO TO 140
C
C     DO A FOURIER TRANSFORM OF LENGTH FOUR (FROM BIT REVERSED ORDER)
C
 50   IP3=IP2*4
      THETA=TWOPI/     (IP3/IP1)
      SINTH= SIN(THETA/2.)
      WSTPR=-2.*SINTH*SINTH
C     COS(THETA)-1, FOR ACCURACY.
      WSTPI= SIN(THETA)
      WR=1.
      WI=0.
      DO 130 I2=1,IP2,IP1
        IF (I2-1) 70,70,60
 60     W2R=WR*WR-WI*WI
        W2I=2.*WR*WI
        W3R=W2R*WR-W2I*WI
        W3I=W2R*WI+W2I*WR
 70     I1MAX=I2+IP1-IP0
        DO 120 I1=I2,I1MAX,IP0
          DO 120 I5=I1,IP5,IP3
            J0=I5
            J1=J0+IP2
            J2=J1+IP2
            J3=J2+IP2
```

```
          IF (I2-1) 90,90,80
C
C         APPLY THE PHASE SHIFT FACTORS
C
 80       TEMPR=DATA(J1)
          DATA(J1)=W2R*TEMPR-W2I*DATA(J1+1)
          DATA(J1+1)=W2R*DATA(J1+1)+W2I*TEMPR
          TEMPR=DATA(J2)
          DATA(J2)=WR*TEMPR-WI*DATA(J2+1)
          DATA(J2+1)=WR*DATA(J2+1)+WI*TEMPR
          TEMPR=DATA(J3)
          DATA(J3)=W3R*TEMPR-W3I*DATA(J3+1)
          DATA(J3+1)=W3R*DATA(J3+1)+W3I*TEMPR
 90       T0R=DATA(J0)+DATA(J1)
          T0I=DATA(J0+1)+DATA(J1+1)
          T1R=DATA(J0)-DATA(J1)
          T1I=DATA(J0+1)-DATA(J1+1)
          T2R=DATA(J2)+DATA(J3)
          T2I=DATA(J2+1)+DATA(J3+1)
          T3R=DATA(J2)-DATA(J3)
          T3I=DATA(J2+1)-DATA(J3+1)
          DATA(J0)=T0R+T2R
          DATA(J0+1)=T0I+T2I
          DATA(J2)=T0R-T2R
          DATA(J2+1)=T0I-T2I
          IF (ISIGN) 100,100,110
 100      T3R=-T3R
          T3I=-T3I
 110      DATA(J1)=T1R-T3I
          DATA(J1+1)=T1I+T3R
          DATA(J3)=T1R+T3I
 120    DATA(J3+1)=T1I-T3R
        TEMPR=WR
        WR=WSTPR*TEMPR-WSTPI*WI+TEMPR
 130  WI=WSTPR*WI+WSTPI*TEMPR+WI
 140  IP2=IP3
      IF (IP3-IP4) 50,150,150
 150  RETURN
      END
C
C
      SUBROUTINE FIXRL (DATA,N,NREM,ISIGN,IFORM)
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
C                                                                   C
C     Subroutine FIXRL is part of the FFT program.  It calculates   C
C     the Fourier transform of length N by the Cooley-Tukey         C
C     algorithm.  The bits are reversed to normal order.            C
C                                                                   C
C     FOR IFORM = 0, CONVERT THE TRANSFORM OF A DOUBLED-UP REAL ARRAY,C
C     CONSIDERED COMPLEX, INTO ITS TRUE TRANSFORM.  SUPPLY ONLY THE  C
C     FIRST HALF OF THE COMPLEX TRANSFORM, AS THE SECOND HALF HAS    C
C     CONJUGATE SYMMETRY.  FOR IFORM = -1, CONVERT THE FIRST HALF    C
C     OF THE TRUE TRANSFORM INTO THE TRANSFORM OF A DOUBLED-UP REAL  C
C     ARRAY.  N MUST BE EVEN.                                        C
C     USING COMPLEX NOTATION AND SUBSCRIPTS STARTING AT ZERO, THE    C
C     TRANSFORMATION IS--                                            C
C     DIMENSION DATA(N,NREM)                                         C
C     ZSTP = EXP(ISIGN*2*PI*I/N)                                     C
C     DO 10 I2=0,NREM-1                                              C
C     DATA(0,I2) = CONJ(DATA(0,I2))*(1+I)                            C
C     DO 10 I1=1,N/4                                                 C
C     Z = (1+(2*IFORM+1)*I*ZSTP**I1)/2                               C
C     I1CNJ = N/2-I1                                                 C
C     DIF = DATA(I1,I2)-CONJ(DATA(I1CNJ,I2))                         C
C     TEMP = Z*DIF                                                   C
C     DATA(I1,I2) = (DATA(I1,I2)-TEMP)*(1-IFORM)                     C
C 10  DATA(I1CNJ,I2) = (DATA(I1CNJ,I2)+CONJ(TEMP))*(1-IFORM)         C
C     IF I1=I1CNJ, THE CALCULATION FOR THAT VALUE COLLAPSES INTO     C
C     A SIMPLE CONJUGATION OF DATA(I1,I2).                           C
```

```
C                                                                       C
C     Subroutine FIXRL is called by:  FOUR2                             C
C                                                                       C
C     Subroutine FIXRL makes no external calls.                         C
C                                                                       C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
      DIMENSION DATA(1)
      DOUBLE PRECISION ZR,ZI,ZSTPR,ZSTPI,SINTH,THETA,TWOPI
      TWOPI=2.*(4.*ATAN(1.))*ISIGN
      IP0=2
      IP1=IP0*(N/2)
      IP2=IP1*NREM
      IF (IFORM) 10,70,70
C
C     PACK THE REAL INPUT VALUES (TWO PER COLUMN)
C
 10   J1=IP1+1
      III2=2
      DATA(III2)=DATA(J1)
      IF (NREM-1) 70,70,20
 20   J1=J1+IP0
      I2MIN=IP1+1
      DO 60 I2=I2MIN,IP2,IP1
        DATA(I2)=DATA(J1)
        J1=J1+IP0
        IF (N-2) 50,50,30
 30     I1MIN=I2+IP0
        I1MAX=I2+IP1-IP0
        DO 40 I1=I1MIN,I1MAX,IP0
          DATA(I1)=DATA(J1)
          DATA(I1+1)=DATA(J1+1)
 40     J1=J1+IP0
 50     DATA(I2+1)=DATA(J1)
 60   J1=J1+IP0
 70   DO 80 I2=1,IP2,IP1
        TEMPR=DATA(I2)
        DATA(I2)=DATA(I2)+DATA(I2+1)
 80   DATA(I2+1)=TEMPR-DATA(I2+1)
      IF (N-2) 200,200,90
 90   THETA=TWOPI/FLOAT(N)
      SINTH= DSIN(THETA/2.)
      ZSTPR=-2.*SINTH*SINTH
      ZSTPI= DSIN(THETA)
      ZR=(1.-ZSTPI)/2.
      ZI=(1.+ZSTPR)/2.
      IF (IFORM) 100,110,110
100   ZR=1.-ZR
      ZI=-ZI
110   I1MIN=IP0+1
      I1MAX=IP0*(N/4)+1
      DO 190 I1=I1MIN,I1MAX,IP0
        DO 180 I2=I1,IP2,IP1
          I2CNJ=IP0*(N/2+1)-2*I1+I2
          IF (I2-I2CNJ) 150,120,120
120       IF (ISIGN*(2*IFORM+1)) 130,140,140
130       DATA(I2+1)=-DATA(I2+1)
140       IF (IFORM) 170,180,180
150       DIFR=DATA(I2)-DATA(I2CNJ)
          DIFI=DATA(I2+1)+DATA(I2CNJ+1)
          TEMPR=DIFR*ZR-DIFI*ZI
          TEMPI=DIFR*ZI+DIFI*ZR
          DATA(I2)=DATA(I2)-TEMPR
          DATA(I2+1)=DATA(I2+1)-TEMPI
          DATA(I2CNJ)=DATA(I2CNJ)+TEMPR
          DATA(I2CNJ+1)=DATA(I2CNJ+1)-TEMPI
          IF (IFORM) 160,180,180
160       DATA(I2CNJ)=DATA(I2CNJ)+DATA(I2CNJ)
          DATA(I2CNJ+1)=DATA(I2CNJ+1)+DATA(I2CNJ+1)
170       DATA(I2)=DATA(I2)+DATA(I2)
```

```
          DATA(I2+1)=DATA(I2+1)+DATA(I2+1)
  180     CONTINUE
          TEMPR=ZR-.5
          ZR=ZSTPR*TEMPR-ZSTPI*ZI+ZR
  190   ZI=ZSTPR*ZI+ZSTPI*TEMPR+ZI
C
C       RECURSION SAVES TIME, AT A SLIGHT LOSS IN ACCURACY.  IF AVAILABLE,
C       USE DOUBLE PRECISION TO COMPUTE ZR AND ZI.
C
  200   IF (IFORM) 270,210,210
C       UNPACK THE REAL TRANSFORM VALUES (TWO PER COLUMN)
  210   I2=IP2+1
        I1=I2
        J1=IP0*(N/2+1)*NREM+1
        GO TO 250
  220   DATA(J1)=DATA(I1)
        DATA(J1+1)=DATA(I1+1)
        I1=I1-IP0
        J1=J1-IP0
  230   IF (I2-I1) 220,240,240
  240   DATA(J1)=DATA(I1)
        DATA(J1+1)=0.
  250   I2=I2-IP1
        J1=J1-IP0
        DATA(J1)=DATA(I2+1)
        DATA(J1+1)=0.
        I1=I1-IP0
        J1=J1-IP0
        IF (I2-1) 260,260,230
  260   III2=2
        DATA(III2)=0.
  270   RETURN
        END
```

APPENDIX D:
SAMPLE INPUT FILE

D1. The following text is an example input file for use with *vib3*. The set of parameters described does not necessarily produce accurate results. Rather, it was chosen for its short length and the size of the output file that it produced. The corresponding output file is listed in Appendix E.

```
Homogeneous system (Model 1)
4 by 10 mesh, square normalized load (5 by 5)
August 20, 1992
1     0    0    0
149  40    1
100000
1 15.65  256   13
3.00
0.100E-04
     1      0.00000      0.00000     1     0
     2    125.00000      0.00000     0     0
     3    250.00000      0.00000     0     0
     4    375.00000      0.00000     0     0
     5    500.00000      0.00000     0     0
     6    625.00000      0.00000     0     0
     7    750.00000      0.00000     0     0
     8    875.00000      0.00000     0     0
     9   1000.00000      0.00000     0     0
    10   1125.00000      0.00000     0     0
    11   1250.00000      0.00000     0     0
    12   1375.00000      0.00000     0     0
    13   1500.00000      0.00000     0     0
    14   1625.00000      0.00000     0     0
    15   1750.00000      0.00000     0     0
    16   1875.00000      0.00000     0     0
    17   2000.00000      0.00000     0     0
    18   2125.00000      0.00000     0     0
    19   2250.00000      0.00000     0     0
    20   2375.00000      0.00000     0     0
    21   2500.00000      0.00000     0     0
    22      0.00000    125.00000     1     0
    23    250.00000    125.00000     0     0
    24    500.00000    125.00000     0     0
    25    750.00000    125.00000     0     0
    26   1000.00000    125.00000     0     0
    27   1250.00000    125.00000     0     0
    28   1500.00000    125.00000     0     0
    29   1750.00000    125.00000     0     0
    30   2000.00000    125.00000     0     0
    31   2250.00000    125.00000     0     0
    32   2500.00000    125.00000     0     0
    33      0.00000    250.00000     1     0
    34    125.00000    250.00000     0     0
    35    250.00000    250.00000     0     0
    36    375.00000    250.00000     0     0
    37    500.00000    250.00000     0     0
    38    625.00000    250.00000     0     0
```

39	750.00000	250.00000	0	0
40	875.00000	250.00000	0	0
41	1000.00000	250.00000	0	0
42	1125.00000	250.00000	0	0
43	1250.00000	250.00000	0	0
44	1375.00000	250.00000	0	0
45	1500.00000	250.00000	0	0
46	1625.00000	250.00000	0	0
47	1750.00000	250.00000	0	0
48	1875.00000	250.00000	0	0
49	2000.00000	250.00000	0	0
50	2125.00000	250.00000	0	0
51	2250.00000	250.00000	0	0
52	2375.00000	250.00000	0	0
53	2500.00000	250.00000	0	0
54	0.00000	375.00000	1	0
55	250.00000	375.00000	0	0
56	500.00000	375.00000	0	0
57	750.00000	375.00000	0	0
58	1000.00000	375.00000	0	0
59	1250.00000	375.00000	0	0
60	1500.00000	375.00000	0	0
61	1750.00000	375.00000	0	0
62	2000.00000	375.00000	0	0
63	2250.00000	375.00000	0	0
64	2500.00000	375.00000	0	0
65	0.00000	500.00000	1	0
66	125.00000	500.00000	0	0
67	250.00000	500.00000	0	0
68	375.00000	500.00000	0	0
69	500.00000	500.00000	0	0
70	625.00000	500.00000	0	0
71	750.00000	500.00000	0	0
72	875.00000	500.00000	0	0
73	1000.00000	500.00000	0	0
74	1125.00000	500.00000	0	0
75	1250.00000	500.00000	0	0
76	1375.00000	500.00000	0	0
77	1500.00000	500.00000	0	0
78	1625.00000	500.00000	0	0
79	1750.00000	500.00000	0	0
80	1875.00000	500.00000	0	0
81	2000.00000	500.00000	0	0
82	2125.00000	500.00000	0	0
83	2250.00000	500.00000	0	0
84	2375.00000	500.00000	0	0
85	2500.00000	500.00000	0	0
86	0.00000	625.00000	1	0
87	250.00000	625.00000	0	0
88	500.00000	625.00000	0	0
89	750.00000	625.00000	0	0
90	1000.00000	625.00000	0	0
91	1250.00000	625.00000	0	0
92	1500.00000	625.00000	0	0

93	1750.00000	625.00000	0	0
94	2000.00000	625.00000	0	0
95	2250.00000	625.00000	0	0
96	2500.00000	625.00000	0	0
97	0.00000	750.00000	1	0
98	125.00000	750.00000	0	0
99	250.00000	750.00000	0	0
100	375.00000	750.00000	0	0
101	500.00000	750.00000	0	0
102	625.00000	750.00000	0	0
103	750.00000	750.00000	0	0
104	875.00000	750.00000	0	0
105	1000.00000	750.00000	0	0
106	1125.00000	750.00000	0	0
107	1250.00000	750.00000	0	0
108	1375.00000	750.00000	0	0
109	1500.00000	750.00000	0	0
110	1625.00000	750.00000	0	0
111	1750.00000	750.00000	0	0
112	1875.00000	750.00000	0	0
113	2000.00000	750.00000	0	0
114	2125.00000	750.00000	0	0
115	2250.00000	750.00000	0	0
116	2375.00000	750.00000	0	0
117	2500.00000	750.00000	0	0
118	0.00000	875.00000	1	0
119	250.00000	875.00000	0	0
120	500.00000	875.00000	0	0
121	750.00000	875.00000	0	0
122	1000.00000	875.00000	0	0
123	1250.00000	875.00000	0	0
124	1500.00000	875.00000	0	0
125	1750.00000	875.00000	0	0
126	2000.00000	875.00000	0	0
127	2250.00000	875.00000	0	0
128	2500.00000	875.00000	0	0
129	0.00000	1000.00000	1	1
130	125.00000	1000.00000	1	1
131	250.00000	1000.00000	1	1
132	375.00000	1000.00000	1	1
133	500.00000	1000.00000	1	1
134	625.00000	1000.00000	1	1
135	750.00000	1000.00000	1	1
136	875.00000	1000.00000	1	1
137	1000.00000	1000.00000	1	1
138	1125.00000	1000.00000	1	1
139	1250.00000	1000.00000	1	1
140	1375.00000	1000.00000	1	1
141	1500.00000	1000.00000	1	1
142	1625.00000	1000.00000	1	1
143	1750.00000	1000.00000	1	1
144	1875.00000	1000.00000	1	1
145	2000.00000	1000.00000	1	1
146	2125.00000	1000.00000	1	1

```
   147   2250.00000   1000.00000      1      1
   148   2375.00000   1000.00000      1      1
   149   2500.00000   1000.00000      1      1
 1   3   1
 2   3   0
 3   3   0
 4   3   0
 5   3   0
 6   3   0
 7   3   0
 8   3   0
 9   3   0
10   3   0
11   3   0
12   3   0
13   3   0
10   4
       1   1     1    22    33     2    34     3    23    35
       2   1     3    23    35     4    36     5    24    37
       3   1     5    24    37     6    38     7    25    39
       4   1     7    25    39     8    40     9    26    41
       5   1     9    26    41    10    42    11    27    43
       6   1    11    27    43    12    44    13    28    45
       7   1    13    28    45    14    46    15    29    47
       8   1    15    29    47    16    48    17    30    49
       9   1    17    30    49    18    50    19    31    51
      10   1    19    31    51    20    52    21    32    53
      11   1    33    54    65    34    66    35    55    67
      12   1    35    55    67    36    68    37    56    69
      13   1    37    56    69    38    70    39    57    71
      14   1    39    57    71    40    72    41    58    73
      15   1    41    58    73    42    74    43    59    75
      16   1    43    59    75    44    76    45    60    77
      17   1    45    60    77    46    78    47    61    79
      18   1    47    61    79    48    80    49    62    81
      19   1    49    62    81    50    82    51    63    83
      20   1    51    63    83    52    84    53    64    85
      21   1    65    86    97    66    98    67    87    99
      22   1    67    87    99    68   100    69    88   101
      23   1    69    88   101    70   102    71    89   103
      24   1    71    89   103    72   104    73    90   105
      25   1    73    90   105    74   106    75    91   107
      26   1    75    91   107    76   108    77    92   109
      27   1    77    92   109    78   110    79    93   111
      28   1    79    93   111    80   112    81    94   113
      29   1    81    94   113    82   114    83    95   115
      30   1    83    95   115    84   116    85    96   117
      31   1    97   118   129    98   130    99   119   131
      32   1    99   119   131   100   132   101   120   133
      33   1   101   120   133   102   134   103   121   135
      34   1   103   121   135   104   136   105   122   137
      35   1   105   122   137   106   138   107   123   139
      36   1   107   123   139   108   140   109   124   141
      37   1   109   124   141   110   142   111   125   143
```

```
    38   1      111    125    143    112    144    113    126    145
    39   1      113    126    145    114    146    115    127    147
    40   1      115    127    147    116    148    117    128    149
1   4.0E06  0.40  0.02  4.
0.2   2.5
0.    2.5
```

APPENDIX E:
SAMPLE OUTPUT FILE

E1. The following text is an example output file produced by *vib3* using the input file listed in Appendix D. The results are not particularly accurate but shown how the data are presented using the minimal output options. The distribution of vertical displacements in the y-direction from the center of the load are shown in Figure E-1 along with the Green's function solution.

```
****************** PROGRAM VIB3 ********************

  This program was written to solve for dynamic displace-
 ments in complex soil/geologic media using a 2-D finite
 element formulation.  The formulation assumes planar
 geometry and material properties in the out-of-plane
 direction and a harmonic source acting on the surface.

  This program was written by David Sykora, at US Army
 Engineer Waterways Experiment Station (WES), Vicksburg,
 MS, under sponsorship of ILIR program (FY90-92).

  Prof. Jose Roesset, Univ. of Texas at Austin, developed
 the condensation procedure used in the formulation as
 successfully implemented by Dr. Kang (1990) for pave-
 ment systems.  Solver subroutines, the FFT routines, and
 the basic framework of the finite element program were
 obtained from Profs. Roesset and Tassoulas, UT.

 THIS SOFTWARE IS DISTRIBUTED AS IS AND WITHOUT WARRANTY
 AS TO PERFORMANCE.  THE USER MUST ASSUME THE RISK OF
 USING THIS SOFTWARE!


 Homogeneous system (Model 1)
 4 by 10 mesh, square normalized load (5 by 5)
 August 20, 1992


*****GENERAL PARAMETERS:

 NUMBER OF TERMS FOR FFT:         256
 INCREMENT OF Y (DY):           15.65
 NUMBER OF MATERIAL TYPES:          1
 NUMBER OF NODES OF INTEREST:       1
 "BIG":                   .1000000E+51
 MAXIMUM ARRAY ALLOCATION:     100000
```

*****PARAMETERS FOR 3-D MESH:

NUMBER OF DIMENSIONS:	3
NUMBER OF ELEMENTS:	40
NUMBER OF NODES:	298
NUMBER OF NODES/ELEMENT:	16
DEGREES OF FREEDOM/NODE:	3
DEGREES OF FREEDOM/ELEMENT:	48

*****PARAMETERS FOR CONDENSED MESH:

NUMBER OF DIMENSIONS:	3
NUMBER OF ELEMENTS:	40
NUMBER OF NODES:	149
NUMBER OF NODES/ELEMENT:	8
DEGREES OF FREEDOM/NODE:	3
DEGREES OF FREEDOM/ELEMENT:	24

FREQUENCIES OF INTEREST (Hz):

3.000 0.000 0.000 0.000 0.000 0.000

NODES OF INTEREST:

1 0 0 0 0 0 0 0

*****MATERIAL PROPERTIES :

MAT	SHEAR MODULUS	POISSONS RATIO	DAMPING RATIO	MASS DENSITY
***	**********	*******	*******	*******
1	0.40000E+07	0.40	0.02	4.00

*****LOADS:

SUBROUTINE YLOAD ASSUMES THAT A NORMALIZED LOAD IS BEING USED!

Left-most extent of load in x-direction = 0.00
Right-most extent of load in x-direction = 2.50

DISTRIBUTED LOAD

*****OUTPUT (DISPLACEMENTS)

CALCULATED AMPLITUDES AT: 3.00 Hz

AMPLITUDES CORRESPONDING TO NODE: 1
(X= 0.000E+00) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	0.154E-08	-0.475E-09	0.161E-08	-0.171E+02
0.156E+02	0.978E-09	-0.445E-09	0.107E-08	-0.245E+02
0.313E+02	0.619E-09	-0.409E-09	0.742E-09	-0.335E+02
0.469E+02	0.378E-09	-0.367E-09	0.526E-09	-0.442E+02
0.626E+02	0.210E-09	-0.318E-09	0.381E-09	-0.566E+02
0.782E+02	0.922E-10	-0.264E-09	0.279E-09	-0.707E+02
0.939E+02	0.118E-10	-0.207E-09	0.208E-09	-0.867E+02
0.110E+03	-0.401E-10	-0.151E-09	0.156E-09	-0.105E+03
0.125E+03	-0.696E-10	-0.969E-10	0.119E-09	-0.126E+03
0.141E+03	-0.812E-10	-0.478E-10	0.943E-10	-0.150E+03
0.156E+03	-0.791E-10	-0.557E-11	0.793E-10	-0.176E+03
0.172E+03	-0.667E-10	0.284E-10	0.725E-10	0.157E+03
0.188E+03	-0.476E-10	0.531E-10	0.713E-10	0.132E+03
0.203E+03	-0.247E-10	0.684E-10	0.728E-10	0.110E+03
0.219E+03	-0.832E-12	0.746E-10	0.746E-10	0.906E+02
0.235E+03	0.217E-10	0.727E-10	0.758E-10	0.733E+02
0.250E+03	0.412E-10	0.638E-10	0.759E-10	0.572E+02
0.266E+03	0.562E-10	0.498E-10	0.751E-10	0.415E+02
0.282E+03	0.661E-10	0.324E-10	0.736E-10	0.261E+02
0.297E+03	0.706E-10	0.136E-10	0.719E-10	0.109E+02
0.313E+03	0.701E-10	-0.481E-11	0.703E-10	-0.393E+01
0.329E+03	0.653E-10	-0.213E-10	0.687E-10	-0.181E+02
0.344E+03	0.572E-10	-0.347E-10	0.669E-10	-0.313E+02
0.360E+03	0.469E-10	-0.442E-10	0.645E-10	-0.433E+02
0.376E+03	0.357E-10	-0.493E-10	0.609E-10	-0.541E+02
0.391E+03	0.249E-10	-0.502E-10	0.560E-10	-0.637E+02
0.407E+03	0.153E-10	-0.472E-10	0.496E-10	-0.721E+02
0.423E+03	0.780E-11	-0.412E-10	0.419E-10	-0.793E+02
0.438E+03	0.288E-11	-0.331E-10	0.332E-10	-0.850E+02
0.454E+03	0.677E-12	-0.239E-10	0.240E-10	-0.884E+02
0.469E+03	0.105E-11	-0.150E-10	0.150E-10	-0.860E+02
0.485E+03	0.358E-11	-0.716E-11	0.800E-11	-0.635E+02
0.501E+03	0.761E-11	-0.135E-11	0.773E-11	-0.101E+02
0.516E+03	0.124E-10	0.186E-11	0.125E-10	0.856E+01
0.532E+03	0.171E-10	0.220E-11	0.172E-10	0.736E+01
0.548E+03	0.209E-10	-0.313E-12	0.209E-10	-0.859E+00
0.563E+03	0.231E-10	-0.536E-11	0.237E-10	-0.131E+02
0.579E+03	0.233E-10	-0.123E-10	0.264E-10	-0.279E+02
0.595E+03	0.212E-10	-0.205E-10	0.295E-10	-0.441E+02
0.610E+03	0.167E-10	-0.289E-10	0.334E-10	-0.600E+02
0.626E+03	0.102E-10	-0.367E-10	0.381E-10	-0.745E+02
0.642E+03	0.193E-11	-0.430E-10	0.430E-10	-0.874E+02

0.657E+03	-0.736E-11	-0.470E-10	0.475E-10	-0.989E+02
0.673E+03	-0.170E-10	-0.482E-10	0.511E-10	-0.109E+03
0.689E+03	-0.261E-10	-0.464E-10	0.532E-10	-0.119E+03
0.704E+03	-0.340E-10	-0.415E-10	0.536E-10	-0.129E+03
0.720E+03	-0.400E-10	-0.338E-10	0.523E-10	-0.140E+03
0.736E+03	-0.436E-10	-0.238E-10	0.497E-10	-0.151E+03
0.751E+03	-0.445E-10	-0.123E-10	0.461E-10	-0.165E+03
0.767E+03	-0.426E-10	-0.173E-13	0.426E-10	-0.180E+03
0.783E+03	-0.381E-10	0.121E-10	0.399E-10	0.162E+03
0.798E+03	-0.312E-10	0.231E-10	0.388E-10	0.144E+03
0.814E+03	-0.226E-10	0.322E-10	0.393E-10	0.125E+03
0.829E+03	-0.129E-10	0.388E-10	0.409E-10	0.108E+03
0.845E+03	-0.276E-11	0.425E-10	0.426E-10	0.937E+02
0.861E+03	0.690E-11	0.431E-10	0.436E-10	0.808E+02
0.876E+03	0.157E-10	0.406E-10	0.435E-10	0.688E+02
0.892E+03	0.228E-10	0.354E-10	0.421E-10	0.572E+02
0.908E+03	0.278E-10	0.280E-10	0.395E-10	0.452E+02
0.923E+03	0.306E-10	0.192E-10	0.361E-10	0.321E+02
0.939E+03	0.310E-10	0.963E-11	0.324E-10	0.173E+02
0.955E+03	0.291E-10	0.239E-12	0.291E-10	0.469E+00
0.970E+03	0.254E-10	-0.821E-11	0.267E-10	-0.179E+02
0.986E+03	0.203E-10	-0.150E-10	0.252E-10	-0.366E+02
0.100E+04	0.142E-10	-0.197E-10	0.242E-10	-0.543E+02
0.102E+04	0.771E-11	-0.218E-10	0.231E-10	-0.705E+02
0.103E+04	0.147E-11	-0.214E-10	0.214E-10	-0.861E+02
0.105E+04	-0.404E-11	-0.184E-10	0.189E-10	-0.102E+03
0.106E+04	-0.843E-11	-0.134E-10	0.158E-10	-0.122E+03
0.108E+04	-0.114E-10	-0.670E-11	0.132E-10	-0.150E+03
0.110E+04	-0.128E-10	0.950E-12	0.129E-10	0.176E+03
0.111E+04	-0.127E-10	0.885E-11	0.155E-10	0.145E+03
0.113E+04	-0.113E-10	0.163E-10	0.198E-10	0.125E+03
0.114E+04	-0.863E-11	0.225E-10	0.241E-10	0.111E+03
0.116E+04	-0.522E-11	0.270E-10	0.275E-10	0.101E+03
0.117E+04	-0.140E-11	0.294E-10	0.295E-10	0.927E+02
0.119E+04	0.245E-11	0.294E-10	0.295E-10	0.852E+02
0.121E+04	0.596E-11	0.270E-10	0.277E-10	0.776E+02
0.122E+04	0.884E-11	0.224E-10	0.241E-10	0.685E+02
0.124E+04	0.109E-10	0.159E-10	0.193E-10	0.557E+02
0.125E+04	0.120E-10	0.810E-11	0.145E-10	0.340E+02
0.127E+04	0.122E-10	-0.482E-12	0.122E-10	-0.227E+01
0.128E+04	0.115E-10	-0.915E-11	0.147E-10	-0.386E+02
0.130E+04	0.101E-10	-0.173E-10	0.200E-10	-0.596E+02
0.131E+04	0.830E-11	-0.242E-10	0.256E-10	-0.711E+02
0.133E+04	0.628E-11	-0.296E-10	0.303E-10	-0.780E+02
0.135E+04	0.430E-11	-0.330E-10	0.333E-10	-0.826E+02
0.136E+04	0.257E-11	-0.343E-10	0.344E-10	-0.857E+02
0.138E+04	0.125E-11	-0.335E-10	0.335E-10	-0.879E+02
0.139E+04	0.431E-12	-0.308E-10	0.308E-10	-0.892E+02
0.141E+04	0.115E-12	-0.265E-10	0.265E-10	-0.898E+02
0.142E+04	0.237E-12	-0.210E-10	0.210E-10	-0.894E+02
0.144E+04	0.665E-12	-0.149E-10	0.149E-10	-0.874E+02
0.146E+04	0.122E-11	-0.869E-11	0.878E-11	-0.820E+02
0.147E+04	0.169E-11	-0.285E-11	0.331E-11	-0.594E+02
0.149E+04	0.187E-11	0.218E-11	0.287E-11	0.494E+02

0.150E+04	0.159E-11	0.608E-11	0.628E-11	0.754E+02
0.152E+04	0.713E-12	0.866E-11	0.869E-11	0.853E+02
0.153E+04	-0.813E-12	0.989E-11	0.993E-11	0.947E+02
0.155E+04	-0.297E-11	0.987E-11	0.103E-10	0.107E+03
0.156E+04	-0.567E-11	0.883E-11	0.105E-10	0.123E+03
0.158E+04	-0.873E-11	0.711E-11	0.113E-10	0.141E+03
0.160E+04	-0.119E-10	0.510E-11	0.130E-10	0.157E+03
0.161E+04	-0.150E-10	0.322E-11	0.153E-10	0.168E+03
0.163E+04	-0.176E-10	0.187E-11	0.177E-10	0.174E+03
0.164E+04	-0.195E-10	0.139E-11	0.196E-10	0.176E+03
0.166E+04	-0.206E-10	0.200E-11	0.207E-10	0.174E+03
0.167E+04	-0.205E-10	0.381E-11	0.209E-10	0.169E+03
0.169E+04	-0.194E-10	0.679E-11	0.205E-10	0.161E+03
0.171E+04	-0.171E-10	0.107E-10	0.202E-10	0.148E+03
0.172E+04	-0.139E-10	0.154E-10	0.207E-10	0.132E+03
0.174E+04	-0.981E-11	0.203E-10	0.225E-10	0.116E+03
0.175E+04	-0.525E-11	0.250E 10	0.255E-10	0.102E+03
0.177E+04	-0.510E-12	0.289E-10	0.290E-10	0.910E+02
0.178E+04	0.408E-11	0.317E-10	0.320E-10	0.827E+02
0.180E+04	0 818E-11	0.328E-10	0.338E-10	0.760E+02
0.182E+04	0.115E-10	0.320E-10	0.340E-10	0.702E+02
0.183E+04	0.138E-10	0.291E-10	0.322E-10	0.645E+02
0.185E+04	0.151E-10	0.240E-10	0.283E-10	0.579E+02
0.186E+04	0.151E-10	0.170E-10	0.227E-10	0.483E+02
0.188E+04	0.141E-10	0.829E-11	0.163E-10	0.305E+02
0.189E+04	0.121E-10	-0.156E-11	0.122E-10	-0.733E+01
0.191E+04	0.945E-11	-0.120E-10	0.153E-10	0.518E+02
0.192E+04	0.642E-11	-0.224E-10	0.233E-10	-0.740E+02
0.194E+04	0.334E-11	-0.320E-10	0.322E-10	-0.840E+02
0.196E+04	0.530E-12	-0.403E-10	0.403E-10	-0.892E+02
0.197E+04	-0.171E-11	-0.466E-10	0.467E-10	-0.921E+02
0.199E+04	-0.316E-11	-0.506E-10	0.507E-10	-0.936E+02
0.200E+04	-0.365E-11	-0.520E-10	0.521E-10	-0.940E+02

AMPLITUDES CORRESPONDING TO NODE: 2
(X= 0.125E+03) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	-0.794E-10	-0.123E-09	0.146E-09	-0.123E+03

AMPLITUDES CORRESPONDING TO NODE: 3
(X= 0.250E+03) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	-0.629E-10	0.120E-09	0.136E-09	0.118E+03

```
-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     4
   (X=  0.375E+03 )         (Z=  0.000E+00 )

 VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

       Y       REAL PART  IMAG. PART   MAGNITUDE     PHASE
  0.000E+00   0.363E-10  -0.160E-10   0.397E-10  -0.238E+02


-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     5
   (X=  0.500E+03 )         (Z=  0.000E+00 )

 VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

       Y       REAL PART  IMAG. PART   MAGNITUDE     PHASE
  0.000E+00   0.419E-11  -0.266E-10   0.270E-10  -0.811E+02


-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     6
   (X=  0.625E+03 )         (Z=  0.000E+00 )

 VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

       Y       REAL PART  IMAG. PART   MAGNITUDE     PHASE
  0.000E+00  -0.578E-11   0.448E-10   0.451E-10   0.974E+02


-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     7
   (X=  0.750E+03 )         (Z=  0.000E+00 )

 VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

       Y       REAL PART  IMAG. PART   MAGNITUDE     PHASE
  0.000E+00   0.412E-10  -0.455E-10   0.614E-10  -0.478E+02


-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     8
   (X=  0.875E+03 )         (Z=  0.000E+00 )

 VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

       Y       REAL PART  IMAG. PART   MAGNITUDE     PHASE
  0.000E+00  -0.214E-10  -0.252E-10   0.330E-10  -0.130E+03


-------------------------------------------------
 AMPLITUDES CORRESPONDING TO NODE:     9
   (X=  0.100E+04 )         (Z=  0.000E+00 )
```

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	-0.216E-10	0.758E-10	0.788E-10	0.106E+03

AMPLITUDES CORRESPONDING TO NODE: 10
(X= 0.112E+04) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	0.278E-10	0.432E-11	0.281E-10	0.884E+01

AMPLITUDES CORRESPONDING TO NODE: 11
(X= 0.125E+04) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	0.976E-11	-0.483E-10	0.493E-10	-0.786E+02

AMPLITUDES CORRESPONDING TO NODE: 12
(X= 0.137E+04) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	-0.580E-11	0.781E-11	0.973E-11	0.127E+03

AMPLITUDES CORRESPONDING TO NODE: 13
(X= 0.150E+04) (Z= 0.000E+00)

VARIATION OF VERTICAL (Z) COMPONENTS IN Y DIRECTION:

Y	REAL PART	IMAG. PART	MAGNITUDE	PHASE
0.000E+00	-0.628E-11	-0.575E-11	0.852E-11	-0.138E+03

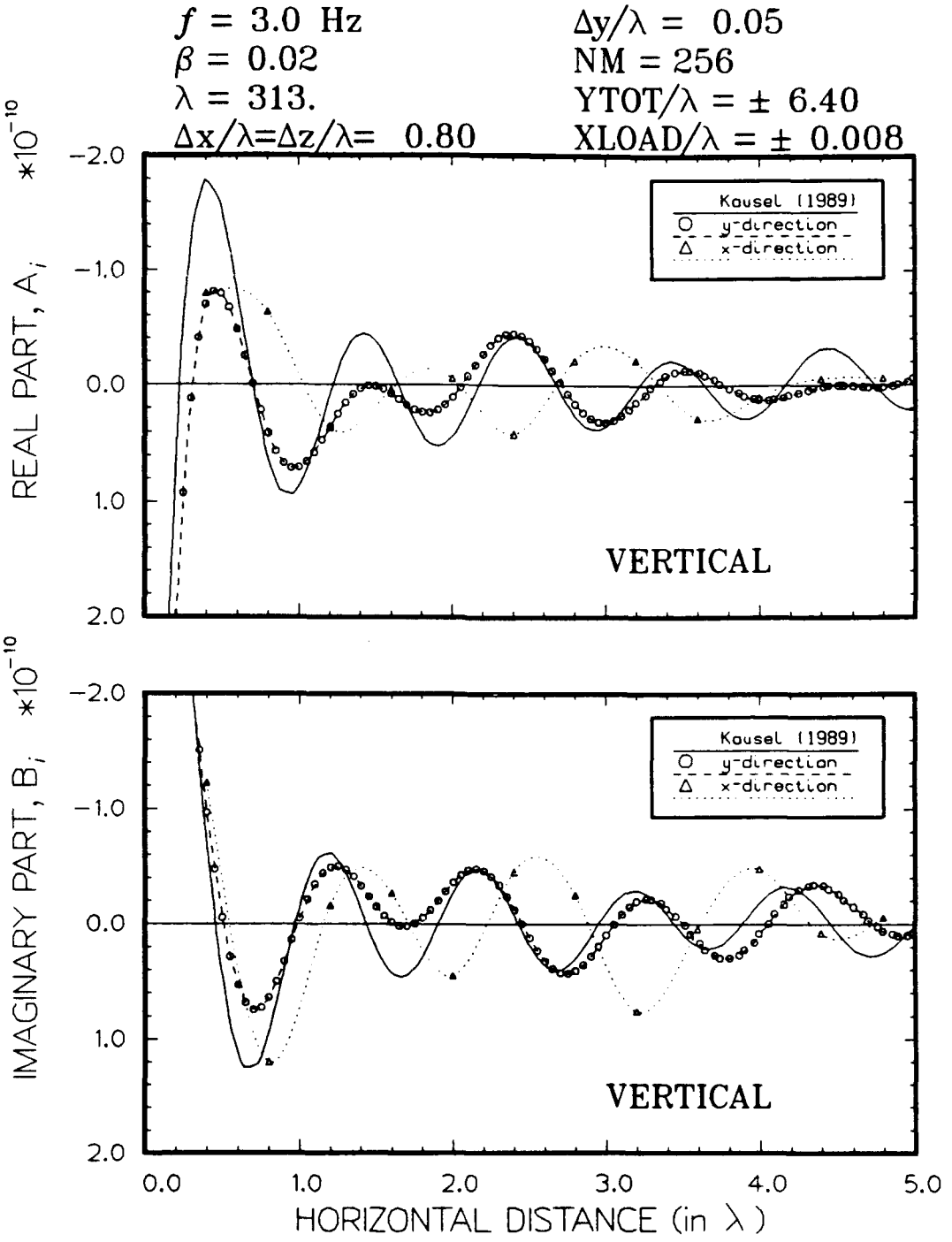


Figure E-1. Variation of dynamic vertical displacements for example problem showing Green's function solutions

Waterways Experiment Station Cataloging-in-Publication Data

Sykora, David W.
Two-dimensional planar geosystems subjected to three-dimensional dynamic loads / by David W. Sykora and Jose M. Roesset ; prepared for Department of the Army, Assistant Secretary of Army (R&D).
191 p. : ill. ; 28 cm. — (Technical report ; GL-92-16)
Includes bibliographic references.
1. Soil dynamics — Data processing. 2. Materials — Dynamic testing. 3. Finite element method. 4. Fourier analysis. I. Roesset, Jose M. II. United States. Assistant Secretary of the Army (R & D) III. U.S. Army Engineer Waterways Experiment Station. IV. Title. V. Series: Technical report (U.S. Army Engineer Waterways Experiment Station) ; GL-92-16.
TA7 W34 no.GL-92-16